

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-099317

(43)Date of publication of application : 07.04.2000

(51)Int.Cl.

G06F 9/06

(21)Application number : 10-265251

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 18.09.1998

(72)Inventor : AMAMIYA MIKA
KAMIO HIROYUKI

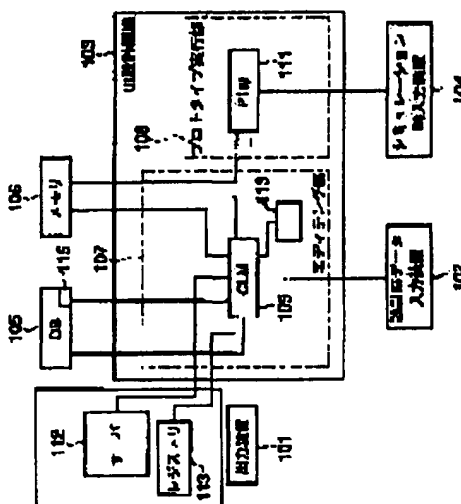
(54) USER INTERFACE DESIGN DEVICE AND METHOD THEREFOR

(57)Abstract:

PROBLEM TO BE SOLVED: To easily design the image of a GUI(graphical user interface) operation screen and also an operation via a GUI operation and by means of a general-purpose software parts.

SOLUTION: The index information on the attributes of software parts, events and messages are read and these index information are stored in a storage means. The optional names are set to these attributes, events and messages respectively (an item definition file 115).

Based on the stored index information and set names of the attributes, events and messages, a list of attributes, events and messages which can be set to the software parts of both linking and linked sides is produced. Based on this list, the attributes, events and messages are set to the software parts of both linking and linked sides.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

BEST AVAILABLE COPY

Copyright (C); 1998,2003 Japan Patent Office

*** NOTICES ***

JPO and NCIP are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.*** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

CLAIMS**[Claim(s)]**

[Claim 1] Arrange various software components in the location of the arbitration on the card showing the actuation screen of a user interface, make said software component of arbitration into a linking agency, and other software components or cards are made into a link place. In the user interface design equipment which designs actuation by matching the event of this link origin, and the message of this link place, and performs simulation of the designed user interface The attribute and event which said software component has, and a reading means to read the index information about a message, A storage means to memorize the index information read with the aforementioned reading means, The attribute and event which said software component has, and a name setting means to set the name of arbitration as a message, The 1st creation means which creates the list of attributes which can be set as the software components of a link place a linking agency based on the index information memorized by said storage means and the name of the attribute set up with said name setting means, An attribute setting means to set an attribute as the software components of a link place said link origin based on said property list, The 2nd creation means which creates the event list of software components of a linking agency based on the index information memorized by said storage means and the name of the event set up with said name setting means, An event setting means to set up the event chosen based on said event list as an event of the software components of said link origin, The 3rd creation means which creates the message list of the software components of a link place based on the index information memorized by said storage means and the name of the message set up with said name setting means, A message setting means to set up as a message of the software components of said link place which matches the message chosen based on said message list with the event set as the software components of said link origin, User interface design equipment characterized by providing.

[Claim 2] Arrange various software components in the location of the arbitration on the card showing the actuation screen of a user interface, make said software component of arbitration into a linking agency, and other software components or cards are made into a link place. In the user interface design equipment which designs actuation by matching the event of this link origin, and the message of this link place, and performs simulation of the designed user interface The attribute and event which said software component has, and a reading means to read the index information about a message, A storage means to memorize the index information read with the aforementioned reading means, A message creation means to set the data used as an argument value as the message which said software component has, and to create a new message, The 1st creation means which creates the list of attributes which can be set as the software components of a link place a linking agency based on the index information memorized by said storage means, An attribute setting means to set an attribute as the software components of a link place said link origin based on said property list, The 2nd creation means which creates the event list of software components of a linking agency based on the index information memorized by said storage means, An event setting means to set up the event chosen based on said event list as an event of the software components of said link origin, The 3rd creation means which creates the message list of the software components of said link place based on the index

information memorized by said storage means and the new message created with said message creation means. A message setting means to set up as a message of the software components of said link place which matches the message chosen based on said message list with the event set as the software components of said link origin, User interface design equipment characterized by providing.

[Claim 3] Arrange various software components in the location of the arbitration on the card showing the actuation screen of a user interface, make said software component of arbitration into a linking agency, and other software components or cards are made into a link place. In the user interface design equipment which designs actuation by matching the event of this link origin, and the message of this link place, and performs simulation of the designed user interface The attribute and event which said software component has, and a reading means to read the index information about a message, A storage means to memorize the index information read with the aforementioned reading means. The software components of the link origin which can be matched with an argument and this message are set as the message which said software component has. The 1st creation means which creates the list of attributes which can be set as the software components of a link place a linking agency based on the index information memorized by a message creation means to create a new message, and said storage means, An attribute setting means to set an attribute as the software components of a link place said link origin based on said property list, The 2nd creation means which creates the event list of software components of a linking agency based on the index information memorized by said storage means, An event setting means to set up the event chosen based on said event list as an event of the software components of said link origin, The 3rd creation means which creates the message list of the software components of a link place based on the index information memorized by said storage means and the new message created with said message creation means, A message setting means to set up as a message of the software components of said link place which matches the message chosen based on said message list with the event set as the software components of said link origin, In case it provides and a message list is created with said 3rd creation means, only when the software components of said link origin are software components of the link origin set as said new message this -- the user interface design equipment characterized by putting a new message on this message list.

[Claim 4] The argument set as the new message created with said message creation means is user interface design equipment according to claim 2 or 3 characterized by being the attribute of the software components set up as a linking agency.

[Claim 5] Arrange various software components in the location of the arbitration on the card showing the actuation screen of a user interface, make said software component of arbitration into a linking agency, and other software components or cards are made into a link place. In the user interface design equipment which designs actuation by matching the event of this link origin, and the message of this link place, and performs simulation of the designed user interface The attribute and event which said software component has, and a reading means to read the index information about a message, A storage means to memorize the index information read with the aforementioned reading means, A message creation means to create the new message for changing the attribute of said software component, The 1st creation means which creates the list of attributes which can be set as the software components of a link place a linking agency based on the index information memorized by said storage means, An attribute setting means to set an attribute as the software components of a link place said link origin based on said property list, The 2nd creation means which creates the event list of software components of a linking agency based on the index information memorized by said storage means, An event setting means to set up the event chosen based on said event list as an event of the software components of said link origin, The 3rd creation means which creates the message list of the software components of a link place based on the index information memorized by said storage means and the message created with said message creation means, A message setting means to set up as a message of the software components of said link place which matches the message chosen based on said message list with the event set as the software components of said link origin, User interface design equipment characterized by providing.

[Claim 6] Arrange various software components in the location of the arbitration on the card showing the actuation screen of a user interface, make said software component of arbitration into a linking agency, and other software components or cards are made into a link place. In the user interface design approach of designing actuation by matching the event of this link origin, and the message of this link place, and performing simulation of the designed user interface The index information about the attribute, event, and message which said software component has is read. Memorize the read index information concerned for a storage means, and the name of arbitration is set as the attribute, event, and message which said software component has. It is based on the index information memorized by said storage means and said set-up name of an attribute. The list of attributes which can be set as the software components of a link place a linking agency is created. Based on this property list, an attribute is set as the software components of a link place said link origin. Based on the index information memorized by said storage means and said set-up name of an event, the event list of software components of a linking agency is created. The event chosen based on this event list is set up as an event of the software components of said link origin. Based on the index information memorized by said storage means and said set-up name of a message, the message list of the software components of a link place is created. The user interface design approach characterized by setting up as a message of the software components of said link place which matches the message chosen based on this message list with the event set as the software components of said link origin.

[Claim 7] Arrange various software components in the location of the arbitration on the card showing the actuation screen of a user interface, make said software component of arbitration into a linking agency, and other software components or cards are made into a link place. In the user interface design approach of designing actuation by matching the event of this link origin, and the message of this link place, and performing simulation of the designed user interface The index information about the attribute, event, and message which said software component has is read. Memorize the read index information concerned for a storage means, and set the data used as an argument value as the message which said software component has, and a new message is created. Based on the index information memorized by said storage means, the list of attributes which can be set as the software components of a link place a linking agency is created. Based on this property list, an attribute is set as the software components of a link place said link origin. Based on the index information memorized by said storage means, the event list of software components of a linking agency is created. The event chosen based on this event list is set up as an event of the software components of said link origin. Based on the index information memorized by said storage means and said created new message, the message list of the software components of a link place is created. The user interface design approach characterized by setting up as a message of the software components of said link place which matches the message chosen based on this message list with the event set as the software components of said link origin.

[Claim 8] Arrange various software components in the location of the arbitration on the card showing the actuation screen of a user interface, make said software component of arbitration into a linking agency, and other software components or cards are made into a link place. In the user interface design approach of designing actuation by matching the event of this link origin, and the message of this link place, and performing simulation of the designed user interface The index information about the attribute, event, and message which said software component has is read. Memorize the read index information concerned for a storage means, and the software components of the link origin which can be matched with an argument and this message are set as the message which said software component has. Create a new message and it is based on the index information memorized by said storage means. The list of attributes which can be set as the software components of a link place a linking agency is created. Based on this property list, an attribute is set as the software components of a link place said link origin. Based on the index information memorized by said storage means, the event list of software components of a linking agency is created. The event chosen based on this event list is set up as an event of the software components of said link origin. Based on the index information memorized by said

storage means and said created new message, the message list of the software components of a link place is created. It sets up as a message of the software components of said link place which matches the message chosen based on this message list with the event set as the software components of said link origin. the time of being the software components of the link origin where the software components of said link origin were set as said new message, in case said message list is created — this — the user interface design approach characterized by putting a new message on this message list.

[Claim 9] The argument set as said new message is the user interface design approach according to claim 7 or 8 characterized by being the attribute of the software components set up as a linking agency.

[Claim 10] Arrange various software components in the location of the arbitration on the card showing the actuation screen of a user interface, make said software component of arbitration into a linking agency, and other software components or cards are made into a link place. In the user interface design approach of designing actuation by matching the event of this link origin, and the message of this link place, and performing simulation of the designed user interface The index information about the attribute, event, and message which said software component has is read. Memorize the read index information concerned for a storage means, and the new message for changing the attribute of said software component is created. Based on the index information memorized by said storage means, the list of attributes which can be set as the software components of a link place a linking agency is created. Based on this property list, an attribute is set as the software components of a link place said link origin. Based on the index information memorized by said storage means, the event list of software components of a linking agency is created. The event chosen based on this event list is set up as an event of the software components of said link origin. Based on the index information memorized by said storage means and said created new message, the message list of the software components of a link place is created. The user interface design approach characterized by setting up as a message of the software components of said link place which matches the message chosen based on this message list with the event set as the software components of said link origin.

[Translation done.]

*** NOTICES ***

JPO and NCIP are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.*** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to suitable user interface design equipment and a suitable approach to perform the design of the screen for the system using the actuation screen which arranged reception and components for the input from a user as a user interface (UI), and a design of operation using a graphical user interface (GUI), and perform the check of operation by simulation.

[0002]

[Description of the Prior Art] The actuation screen design by arranging components and a design of operation are realized by GUI actuation, and there are many tools performed as a design.

[0003] For example, if VisualBasic (U.S. Microsoft registration goods of a shrine) is used, Windows (U.S. Microsoft registration goods of a shrine) is easily programmable using GUI actuation.

[0004]

[Problem(s) to be Solved by the Invention] however, VisualBasic etc. — while a visual programming tool can build the user interface which performs complicated actuation, in order for the great portion of design of operation to describe a program code and to perform it, a designer needs to master programming.

[0005] Moreover, the attribute was beforehand set as the components dealt with within equipment, and the user interface design equipment indicated by JP,10-21062,A was describing the program combining the attribute to the case index. However, it was impossible for it to have been necessary to hold in equipment beforehand for this attribute, and to have incorporated general-purpose software components to equipment dynamically.

[0006] Furthermore, when general-purpose software components were used for the design of the user interface which has a certain specific purpose, there were the following troubles.

[0007] (1) It is unclear what kind of component it is by the general-purpose name of the software component concerned.

[0008] (2) the repeat of the same activity [associate / in many cases / therefore / the message with an argument same among the software components defined beforehand in case the message which needs an argument is associated among software components] — many.

[0009] (3) In order to design the user interface aiming at the property of general-purpose software components, it cannot change dynamically.

[0010] Then, this invention was made in consideration of the above-mentioned situation, and the purpose has the image design of the GUI actuation screen (user interface actuation screen) using general-purpose software components, and a design of operation in offering the user interface design equipment and the approach of performing easily (not describing a program) by GUI (graphical user interface) actuation (actuation which associates an event and a message).

[0011] Moreover, this invention is to offer the user interface design equipment and the approach of incorporating dynamically the general-purpose software components arranged on a GUI actuation screen in equipment.

[0012]

[Means for Solving the Problem] Claim 1 : the user interface design equipment of this invention Arrange various software components (for example, OCX) in the location of the arbitration on the card showing the actuation screen of a user interface, make said software component of arbitration into a linking agency, and other software components or cards are made into a link place. In the user interface design equipment which designs actuation by matching the event of this link origin, and the message of this link place, and performs simulation of the designed user interface The attribute and event which said software component has, and a reading means to read the index information (property information, event information, message information, argument information) about a message (for OCX information to be read in registry 113), A storage means to memorize the index information read with the aforementioned reading means (table 114), The attribute and event which said software component has, and a name setting means to set the name of arbitration as a message (creation of the item definition file 115), It is based on the index information memorized by said storage means and the name (property Japanese name information) of the attribute set up with said name setting means. The 1st creation means which creates the list of attributes which can be set as the software components of a link place a linking agency (refer to drawing 12), An attribute setting means to set an attribute as the software components of a link place said link origin based on said property list (link setting dialog etc.), It is based on the index information memorized by said storage means and the name (event Japanese name information) of the event set up with said name setting means. The 2nd creation means which creates the event list of software components of a linking agency (refer to drawing 15), An event setting means to set up the event chosen based on said event list as an event of the software components of said link origin (link setting dialog etc.), The 3rd creation means which creates the message list (refer to drawing 16) of the software components of a link place based on the index information memorized by said storage means and the name (message Japanese name information) of the message set up with said name setting means, A message setting means to set up as a message of the software components of said link place which matches the message chosen based on said message list with the event set as the software components of said link origin (link setting dialog etc.), It is characterized by providing.

[0013] According to this invention, the image design of a GUI actuation screen (user interface actuation screen) and a design of operation can carry out easily (not describing a program) by GUI (graphical user interface) actuation using general-purpose software components.

[0014] That is, when under the name of the default of the property which becomes possible [incorporating dynamically the general-purpose software components arranged on a GUI actuation screen in equipment], and these software components have, an event, and a message, and the manager of GUI design equipment attaches the alias name it is easier for a designer to understand, the design environment of a GUI actuation screen improves more.

[0015] Claim 2: Arrange various software components in the location of the arbitration on the card showing the actuation screen of a user interface, make said software component of arbitration into a linking agency, and other software components or cards are made into a link place. In the user interface design equipment which designs actuation by matching the event of this link origin, and the message of this link place, and performs simulation of the designed user interface The attribute and event which said software component has, and a reading means to read the index information about a message, A storage means to memorize the index information read with the aforementioned reading means, A message creation means to set the data used as an argument value as the message which said software component has, and to create a new message (message macro with an argument), The 1st creation means which creates the list of attributes which can be set as the software components of a link place a linking agency based on the index information memorized by said storage means, An attribute setting means to set an attribute as the software components of a link place said link origin based on said property list, The 2nd creation means which creates the event list of software components of a linking agency based on the index information memorized by said storage means, An event setting means to set up the event chosen based on said event list as an event of the software components of said link origin, The 3rd creation means which creates the message list of the software components

of a link place based on the index information memorized by said storage means and the new message created with said message creation means, It is characterized by providing a message setting means to set up as a message of the software components of said link place which matches the message chosen based on said message list with the event set as the software components of said link origin.

[0016] According to this invention, the image design of a GUI actuation screen (user interface actuation screen) and a design of operation can carry out easily (not describing a program) by GUI (graphical user interface) actuation using general-purpose software components.

[0017] That is, since the contents of the data which become possible [incorporate dynamically the general-purpose software components arranged on a GUI actuation screen in equipment], and are sent as an argument can register the same message as a message macro and can reuse it, it makes it possible to omit the time and effort of the link design which sends the message which needs an argument, and the design environment of a GUI actuation screen improves more.

[0018] Claim 3: Arrange various software components in the location of the arbitration on the card showing the actuation screen of a user interface, make said software component of arbitration into a linking agency, and other software components or cards are made into a link place. In the user interface design equipment which designs actuation by matching the event of this link origin, and the message of this link place, and performs simulation of the designed user interface The attribute and event which said software component has, and a reading means to read the index information about a message, A storage means to memorize the index information read with the aforementioned reading means, The software components of the link origin which can be matched with an argument and this message are set as the message which said software component has. A message creation means to create a new message (message macro with an argument), The 1st creation means which creates the list of attributes which can be set as the software components of a link place a linking agency based on the index information memorized by said storage means, An attribute setting means to set an attribute as the software components of a link place said link origin based on said property list, The 2nd creation means which creates the event list of software components of a linking agency based on the index information memorized by said storage means, An event setting means to set up the event chosen based on said event list as an event of the software components of said link origin, The 3rd creation means which creates the message list of the software components of a link place based on the index information memorized by said storage means and the new message created with said message creation means, A message setting means to set up as a message of the software components of said link place which matches the message chosen based on said message list with the event set as the software components of said link origin, the time of being the software components of the link origin where the software components of said link origin were set as said new message, in case it provides and a message list is created with said 3rd creation means — this — it is characterized by putting a new message on this message list.

[0019] According to this invention, the image design of a GUI actuation screen (user interface actuation screen) and a design of operation can carry out easily (not describing a program) by GUI (graphical user interface) actuation using general-purpose software components.

[0020] Namely, it becomes possible to incorporate dynamically the general-purpose software components arranged on a GUI actuation screen in equipment. Moreover, since a linking agency can be limited beforehand in the case of the message with the same contents of the data sent as an argument, and the message which sends the attribute of a linking agency to an argument, it can register as a message macro and it can reuse The time and effort of the link design which sends the message which needs an argument can be omitted, and the design environment of a GUI actuation screen improves more.

[0021] Claim 4: The argument preferably set as the new message created with said message creation means is characterized by being the attribute of the software components set up as a linking agency.

[0022] Claim 5: Arrange various software components in the location of the arbitration on the card showing the actuation screen of a user interface, make said software component of

arbitration into a linking agency, and other software components or cards are made into a link place. In the user interface design equipment which designs actuation by matching the event of this link origin, and the message of this link place, and performs simulation of the designed user interface The attribute and event which said software component has, and a reading means to read the index information about a message, A storage means to memorize the index information read with the aforementioned reading means, A message creation means to create the new message for changing the attribute of said software component, The 1st creation means which creates the list of attributes which can be set as the software components of a link place a linking agency based on the index information memorized by said storage means, An attribute setting means to set an attribute as the software components of a link place said link origin based on said property list, The 2nd creation means which creates the event list of software components of a linking agency based on the index information memorized by said storage means, An event setting means to set up the event chosen based on said event list as an event of the software components of said link origin, The 3rd creation means which creates the message list of the software components of a link place based on the index information memorized by said storage means and the message created with said message creation means, It is characterized by providing a message setting means to set up as a message of the software components of said link place which matches the message chosen based on said message list with the event set as the software components of said link origin.

[0023] According to this invention, the image design of a GUI actuation screen (user interface actuation screen) and a design of operation can carry out easily (not describing a program) by GUI (graphical user interface) actuation using general-purpose software components.

[0024] That is, the message for becoming possible to incorporate dynamically the general-purpose software components arranged on a GUI actuation screen in equipment, and changing the property of software components dynamically at the time of the design of a GUI actuation screen can be used, and the design environment of a GUI actuation screen improves more.

[0025] Claim 6 : the user interface design approach of this invention Arrange various software components in the location of the arbitration on the card showing the actuation screen of a user interface, make said software component of arbitration into a linking agency, and other software components or cards are made into a link place. In the user interface design approach of designing actuation by matching the event of this link origin, and the message of this link place, and performing simulation of the designed user interface The index information about the attribute, event, and message which said software component has is read. Memorize the read index information concerned for a storage means, and the name of arbitration is set as the attribute, event, and message which said software component has. It is based on the index information memorized by said storage means and said set-up name of an attribute. The list of attributes which can be set as the software components of a link place a linking agency is created. Based on this property list, an attribute is set as the software components of a link place said link origin. Based on the index information memorized by said storage means and said set-up name of an event, the event list of software components of a linking agency is created. The event chosen based on this event list is set up as an event of the software components of said link origin. Based on the index information memorized by said storage means and said set-up name of a message, the message list of the software components of a link place is created. It is characterized by setting up as a message of the software components of said link place which matches the message chosen based on this message list with the event set as the software components of said link origin.

[0026] According to invention, the image design of a GUI actuation screen (user interface actuation screen) and a design of operation can carry out easily (not describing a program) by GUI (graphical user interface) actuation using general-purpose software components.

[0027] That is, when under the name of the default of the property which becomes possible [incorporating dynamically the general-purpose software components arranged on a GUI actuation screen in equipment], and these software components have, an event, and a message, and the manager of GUI design equipment attaches the alias name it is easier for a designer to understand, the design environment of a GUI actuation screen improves more.

[0028] Claim 7: The user interface design approach of this invention Arrange various software components in the location of the arbitration on the card showing the actuation screen of a user interface, make said software component of arbitration into a linking agency, and other software components or cards are made into a link place. In the user interface design approach of designing actuation by matching the event of this link origin, and the message of this link place, and performing simulation of the designed user interface The index information about the attribute, event, and message which said software component has is read. Memorize the read index information concerned for a storage means, and set the data used as an argument value as the message which said software component has, and a new message is created. Based on the index information memorized by said storage means, the list of attributes which can be set as the software components of a link place a linking agency is created. Based on this property list, an attribute is set as the software components of a link place said link origin. Based on the index information memorized by said storage means, the event list of software components of a linking agency is created. The event chosen based on this event list is set up as an event of the software components of said link origin. Based on the index information memorized by said storage means and said created new message, the message list of the software components of a link place is created. It is characterized by setting up as a message of the software components of said link place which matches the message chosen based on this message list with the event set as the software components of said link origin.

[0029] According to this invention, the image design of a GUI actuation screen (user interface actuation screen) and a design of operation can carry out easily (not describing a program) by GUI (graphical user interface) actuation using general-purpose software components.

[0030] That is, since the contents of the data which become possible [incorporate dynamically the general-purpose software components arranged on a GUI actuation screen in equipment], and are sent as an argument can register the same message as a message macro and can reuse it, it makes it possible to omit the time and effort of the link design which sends the message which needs an argument, and the design environment of a GUI actuation screen improves more.

[0031] Claim 8: The user interface design approach of this invention Arrange various software components in the location of the arbitration on the card showing the actuation screen of a user interface, make said software component of arbitration into a linking agency, and other software components or cards are made into a link place. In the user interface design approach of designing actuation by matching the event of this link origin, and the message of this link place, and performing simulation of the designed user interface The index information about the attribute, event, and message which said software component has is read. Memorize the read index information concerned for a storage means, and the software components of the link origin which can be matched with an argument and this message are set as the message which said software component has. Create a new message and it is based on the index information memorized by said storage means. The list of attributes which can be set as the software components of a link place a linking agency is created. Based on this property list, an attribute is set as the software components of a link place said link origin. Based on the index information memorized by said storage means, the event list of software components of a linking agency is created. The event chosen based on this event list is set up as an event of the software components of said link origin. Based on the index information memorized by said storage means and said created new message, the message list of the software components of a link place is created. It sets up as a message of the software components of said link place which matches the message chosen based on this message list with the event set as the software components of said link origin. the time of being the software components of the link origin where the software components of said link origin were set as said new message, in case said message list is created — this — it is characterized by putting a new message on this message list.

[0032] According to this invention, the image design of a GUI actuation screen (user interface actuation screen) and a design of operation can carry out easily (not describing a program) by GUI (graphical user interface) actuation using general-purpose software components.

[0033] Namely, it becomes possible to incorporate dynamically the general-purpose software

components arranged on a GUI actuation screen in equipment. Moreover, since a linking agency can be limited beforehand in the case of the message with the same contents of the data sent as an argument, and the message which sends the attribute of a linking agency to an argument, it can register as a message macro and it can reuse The time and effort of the link design which sends the message which needs an argument can be omitted, and the design environment of a GUI actuation screen improves more.

[0034] Claim 9: The argument set as said new message is preferably characterized by being the attribute of the software components set up as a linking agency.

[0035] Claim 10: The user interface design approach of this invention Arrange various software components in the location of the arbitration on the card showing the actuation screen of a user interface, make said software component of arbitration into a linking agency, and other software components or cards are made into a link place. In the user interface design approach of designing actuation by matching the event of this link origin, and the message of this link place, and performing simulation of the designed user interface The index information about the attribute, event, and message which said software component has is read. Memorize the read index information concerned for a storage means, and the new message for changing the attribute of said software component is created. Based on the index information memorized by said storage means, the list of attributes which can be set as the software components of a link place a linking agency is created. Based on this property list, an attribute is set as the software components of a link place said link origin. Based on the index information memorized by said storage means, the event list of software components of a linking agency is created. The event chosen based on this event list is set up as an event of the software components of said link origin. Based on the index information memorized by said storage means and said created new message, the message list of the software components of a link place is created. It is characterized by setting up as a message of the software components of said link place which matches the message chosen based on this message list with the event set as the software components of said link origin.

[0036] According to this invention, the image design of a GUI actuation screen (user interface actuation screen) and a design of operation can carry out easily (not describing a program) by GUI (graphical user interface) actuation using general-purpose software components.

[0037] That is, the message for becoming possible to incorporate dynamically the general-purpose software components arranged on a GUI actuation screen in equipment, and changing the property of software components dynamically at the time of the design of a GUI actuation screen can be used, and the design environment of a GUI actuation screen improves more.

[0038]

[Embodiment of the Invention] Hereafter, the gestalt of operation of this invention is explained with reference to a drawing.

[0039] (1) **** of user interface design equipment -- explain the configuration of user interface design equipment first.

[0040] The (user interface UI) design equipment concerning this operation gestalt The output unit 101 which mainly contains displays, such as a loudspeaker used for the output of voice or a sound (sound) and a CRT display, and a liquid crystal display, as shown in drawing 2 . The input device 102 which consists of input means, such as a keyboard and a mouse, and is used for the input of the data at the time of a design etc. (a data entry unit is hereafter called at the time of a design), The user interface design device 103 in which an interactive dialogue with a designer or a user is performed at the time of a design and simulation (UI design device is called hereafter), A touch panel, a microphone (for voice input), (For a touch input) The input unit 104 which consists of input means, such as a mouse and a keyboard, and is used for the information input at the time of simulation (an input unit is hereafter called at the time of simulation), It consists of memory 106 which offers the field for holding the material of the database (DB) 105 which saves and manages the material of the designed screen information, performance information, and data etc., and the designed screen information, performance information and data etc. temporarily, a working area, etc. Here, an input unit 104 is the same input unit physically at the time of a data entry unit 102 and simulation at the time of a design. The item

definition file 115 mentioned later is stored in a database 105.

[0041] The user interface design equipment concerning this operation gestalt is realized using the personal computer (computer) equipped with large capacity storage, such as output means, such as input means, such as a touch panel, a microphone, a mouse, and a keyboard, a loudspeaker, and a display (indicating equipment), a floppy disk drive unit, and a hard disk drive unit, main memory, CPU, etc.

[0042] Especially the function that UI design device 103 in which the core of the user interface design equipment of drawing 2 is made has equips the floppy disk drive unit in the above-mentioned personal computer with the record medium concerned, and is realized by reading and performing the program currently recorded on the floppy disk concerned with the above-mentioned personal computer (inner CPU) using the storage which recorded the program for performing the function with the above-mentioned personal computer, for example, a floppy disk.

[0043] For example, as shown in drawing 1, the program for performing the function which UI design device 103 recorded on the record medium has is executed on a computer 10, and the display 13 as the keyboard 11 as an input means and a mouse 12, and an output means is connected to the computer 10.

[0044] UI design device 103 consists of the prototype activation sections 108 which manage the management at the time of the simulation for the editing section 107 which manages the management at the time of the design of a screen or actuation, and the check of designed UI of operation. About the configuration of this editing section 107 and the prototype activation section 108, it mentions later.

[0045] (2) Explain the outline of UI design device 103, next the outline of UI design device 103 with reference to the example of a screen image of drawing 3.

[0046] UI design device 103 of drawing 2 is a tool which designs screen use of UI, and operating specification by GUI. The design of the screen by UI design device 103 is performed on the screen displayed on the display 13 as an output unit 101, as shown in drawing 3. That is, the card 202 as a screen object (object as used in the field of object-oriented) which expresses UI actuation screen on the card layout manager window (CLMW) 201 is created, and a screen design is performed by arranging the software components 202 on the card concerned.

[0047] The thing of arbitration is chosen from a tool bar 204, and software components are arranged from it. With this operation gestalt, OCX (OLE (Object Linking and Embedding) custom control) used for example, on Win32 (registration goods of U.S. Microsoft Microsoft Corp.) API (Application Program Interface) is used as this software component.

[0048] It has the global unique identification information which expressed the class of sort wear components called CLSID, respectively to each OCX. Moreover, index information (event information, message information, property information, argument information), such as an attribute which the OCX concerned has, an event, and a message, a method, etc. are described by the file (stereo file) of OCX so that it may mention later besides CLSID.

[0049] It is the server 112 of drawing 2 which supplies OCX. OCX which a user wants to use with the GUI design equipment concerned needs to be beforehand installed in the computer 10 (before starting UI design equipment and starting a design). namely, the computer 10 of drawing 1 -- stores, such as a hard disk, -- a floppy disk, CD-ROM, etc. -- using it -- OCX -- instrumental -- a roll writes the file of the OCX concerned in the predetermined storage region of a computer 10. The storage region where the file of OCX was written in is called a server 112. Although it is necessary to register CLSID of the installed OCX concerned into OS (operation system) in that case, the registry of Windows (registration goods of U.S. Microsoft) is used here, for example.

[0050] Install of OCX registers into registry 113 CLSID of the OCX concerned, the pointer information to the storage region where the identifier (for example, components name, such as a "carbon button" and a "count result display box") of the OCX concerned and the file of OCX are memorized corresponding to it, etc. (it is hereafter called OCX information).

[0051] Install (creation of a server 112, registration to registry 113) of such OCX is automatically performed in this case by the installer corresponding to Windows.

[0052] In addition, at the time of the design by UI design device 103, the map viewer 205 may be displayed with CLMW201. This map viewer 205 is a window for looking through and displaying a card [finishing / creation], and the card icon 206 which is the contraction image of a card [finishing / creation] is displayed on the map viewer 205 concerned. When there are many cards [finishing / creation], only the card icon corresponding to some cards is displayed on the map viewer 205, and the card icon corresponding to the remaining cards is displayed by scrolling to the upper and lower sides or right and left on the map viewer 205 concerned. Moreover, when the link to other cards [components / (it is also hereafter called a components object) / which are arranged on a card / software] is stretched, the arrow head (link line) showing a link and its direction is displayed between corresponding card icons.

[0053] (3) Explain the class of manager (management tool) which the editing section 107 and the prototype activation section 108 in the class of manager, next drawing 2 have.

[0054] First, with generation of a card 202, and management, the editing section 107 in drawing 2 reads software components from a server 112, and consists of card layout managers (CLM) 109 arranged on a card 202.

[0055] CLM109 performs a link setup for edit of the software components arranged on the card 202 and the card 202, and the processing of operation performed to the event (the input with voice which clicks a carbon button is performed) from a user at the time of starting. Moreover, software components (namely, OCX) are read from a server 112, and it has the function as the so-called container registered into a card 202.

[0056] On the other hand, the prototype activation section 108 in drawing 2 is constituted by the prototype manager (PTM) 111. PTM111 is started by CLM109 according to the directions from a user. At the time of starting of this PTM111, the software components which received the event from a user send a message to the set-up link place by that message-sending function. The repeat of this message sending can perform simulation of operation as the design of a link.

[0057] (4) In case a user interface is designed using the user interface design equipment of creation drawing 2 of an item definition file, in order to offer the environment which is easier to use, the user interface design equipment concerned has the function (the addition function of the name of the arbitration to software components, the setting up function of the macro of a message with an argument, setting up function of a property change macro) for customizing OCX which is general-purpose software components.

[0058] Drawing 25 shows an example of the customize window of OCX displayed on an output unit 101 by CLM109. Drawing 25 shows the customize window of OCX whose components name is a "carbon button", for example. For example, a window as shown in drawing 25 is displayed by directing customize from the actuation menu to the OCX concerned displayed by choosing desired OCX from the tool bar 204 on a display screen as shown in drawing 3 R> 3 currently displayed on the output unit 101.

[0059] As shown in drawing 25, the tabs 1301a-1301e for 5 pages are displayed on the customize window of OCX, and the Japanese name of the property (attribute) of OCX, the Japanese name of an event, the Japanese name of a message, the macro of a message with an argument, and the macro of a property change are set to it, respectively on the page 1302 corresponding to each tabs 1301a-1301e.

[0060] The contents set up in this customize window are saved by text format in a database 105 for every OCX as an item definition file 115. The item definition file 115 shall have a file name of "components name .itm of the OCX concerned" about one OCX.

[0061] First, the case where the Japanese name of the arbitration for which a user asks is set as the property which the selected OCX concerned has, an event, and a message is explained.

[0062] In drawing 25, the property Japanese name setting page 1302 for a components name to set the Japanese name of the arbitration for which a user asks to OCX of a "carbon button" as the property which the OCX has is chosen.

[0063] The list of the property names which the OCX concerned chosen has in the field the "property name" of a list box 1303, beforehand is displayed on a vertical single tier. If a desired thing is clicked and chosen among this property name, the text input box 1304 will be displayed.

[0064] If a desired Japanese name is inputted into the text input box 1304 and the input

concerned is decided, the text input box 1304 will disappear and the Japanese name inputted into the field on the right of the selected property name concerned will be displayed.

[0065] An event and the Japanese name of a message also choose and display each page, and should just set it up in the same procedure as the above.

[0066] When a setup on a property, an event, and the Japanese name setting page 1302 of a message is completed, the "O.K." carbon button 1305 is clicked and the setup concerned is made to decide. Then, the property set up on these pages, an event, and the Japanese name of a message are saved at the item definition file of the OCX concerned, for example, "carbon button .itm."

[0067] Next, the case where a message macro with an argument (it may be hereafter called a message macro simply) is set up is explained.

[0068] Drawing 26 is a page which sets up the macro of the message with an argument of OCX of the components name for example, of a count result display box.

[0069] If this OCX is chosen as a link place at the time of a link setup which will be later mentioned if the message macro of a message with an argument is set up, this macro comes to be shown in the message select list of drawing 16 a list table as a message, and it can use as a message to which the argument is already set (even if it does not set up an argument at the time of a link setup). When sending a message by the argument of the same contents by this function, since a link setup can be performed in the form where a setup of an argument was omitted, a designer's time and effort can be saved.

[0070] In drawing 26, the macro name of a message macro [finishing / a setup / already] is shown in the macro name list 1500 the list table. In order to delete the macro whose New button 1501 has been set up to set up a message macro newly, one macro name to delete from the macro name list 1500 is chosen, and the deletion carbon button 1502 is pushed.

[0071] A push on New button 1501 displays the argument message macro setting window 1600 as shown in drawing 27. Here, a desired macro name is inputted, the class (components name) of OCX which becomes a linking agency to the linking agency selection field 1602 is chosen and inputted, and a message is chosen and inputted into the macro name input area 1601 to the message selection field 1603. In case a linking agency and a message are chosen in the linking agency selection field 1602 and the message selection field 1603, the linking agency select list and message select list which the selectable link origin which the selected OCX concerned has beforehand, and a message enumerated are displayed, and as a desired thing is chosen, it may be carried out from the list. At this time, it can also choose "it is nothing" from a linking agency select list a linking agency. When it is not necessary to set the property of a linking agency as an argument, or in enabling it to want to be able to call this message macro from all links origin at the time of a link setup (he wants to limit a linking agency and there is nothing), it chooses "it is nothing" a linking agency. When other, the class of OCX of a linking agency is chosen.

[0072] Then, a push on the argument carbon button 1604 displays the argument list dialog 1100 as shown in drawing 18, for example.

[0073] As shown in drawing 18, the message name chosen previously and the argument selection field 1101 which displays that the list of the arguments which the message concerned has is selectable are displayed on the argument list dialog 1100.

[0074] The list of the arguments displayed on the argument selection field 1101 enumerates the parameter identifiers which are what CLM109 created with reference to the argument list contained in the message information on the message chosen previously, for example, are in an argument list.

[0075] Here, since it has the one argument a "figure" from message information, only the argument a "figure" is shown in the message the "figure input" of the count result display box 701 which is OCX of a link place by the argument selection field 1101 of drawing 18. When it, of course, has two or more arguments, all they are displayed on the argument selection field 1101.

[0076] A desired parameter identifier is clicked and chosen from the argument selection field 1101 of drawing 18, and further, if the depression of the setup key 1102 is carried out, the argument setting dialog 1103 as shown in drawing 19 will be displayed. The argument setting dialog 1103 is a dialog which decides of what kind of value to set an argument as the selected

message concerned.

[0077] As shown in drawing 19, the parameter identifier which CLM109 read from the argument information concerned on a table 114 to each of the parameter identifier display edit box 1104 and the argument mold display edit box 1105, and the mold of an argument are displayed on the argument setting dialog 1103. Here, it turns out that the value of integer type (int) is set as the argument a "figure."

[0078] In case a value is set as the argument concerned, it chooses whether the value of immobilization is sent as an argument with the set point change radio button 1106, or the property name of OCX is sent as a value of an argument. In sending a fixed value as an argument, it inputs a desired value (for example, "0") into the set point input edit box 1107.

[0079] In sending the property (value) of OCX as an argument, it chooses of which software component (OCX) it is a property name. First, the carbon button of facing down in the right-hand side of the components name selection field 1108 by which it is indicated by the arrow head is clicked. Then, although a components name select list is displayed, it cannot choose only two, "a linking agency" and a "link place", here (in addition, when "to be nothing" is chosen in the linking agency by drawing 27, only a "link place" is selectable). It is because it is only the fact of being one-kind) when OCX which this does not know what kind of OCX is created on the screen at the time of a link setup at the time of a message macro setup, but is surely created as known information at the time of a link setup has two kinds of OCX specified as the link place the linking agency of the same OCX of (link dimension and a link place.

[0080] The carbon button of facing down which chooses "a linking agency" by the components name select list, next is in the right-hand side of the property name selection field 1109 by which it is indicated by the arrow head is clicked. Then, a property name select list is displayed. The property name of OCX chosen in the linking agency selection field 1601 of drawing 27 is shown in a property name select list a list table. The property name of OCX under current selection (it is under customize) is shown in the property select list displayed when a "link place" is chosen by the components name select list a list table. What is necessary is to click a desired property name and just to choose from this property name select list.

[0081] In addition, a property name select list reads the property information on the components which CLM109 chose from the components name select list from a table 114, and is displayed as a pull down menu which extracted the property name of the same mold as the mold currently displayed on the argument mold display edit box 1105 among the property names which the selected component concerned has, and indicated by the list.

[0082] A setup of an argument is completed by carrying out the depression of the O.K. carbon button 1101 to the last.

[0083] Here, the case where the message macro of a message "Insert Number" with one argument named "Number" of OCX called a count result display box is created is taken for an example, and it explains more concretely. The macro name "count" is inputted into the macro name input area 1601 of the message macro setting window 1600 of drawing 27 R> 7, OCX a "carbon button" is chosen as it as a linking agency in the linking agency selection field 1602, and the message "Insert Number" is chosen in the message selection field 1603.

[0084] Furthermore, the argument carbon button 1804 is pushed and a parameter identifier "Number" is chosen in the argument list dialog 1100, and subsequently to an argument, in the argument setting dialog 1103, it sets up so that a property may be sent with the set point change radio button 1106. That is, if "a linking agency" is chosen in the components name selection field 1108, in the property name selection field 1109, a property name select list is displayed, and by **, the property of OCX the "carbon button" chosen as a linking agency in the linking agency selection field 1602 will choose from there the property "Data", and will push the "O.K." carbon button 1100. Thus, "count" message macro of OCX called the created count result display box can be used as a message, when performing a link setup to OCX called a count result display box from OCX called a carbon button.

[0085] The link set up using "count" message macro is changed into the code which calls the "Insert Number" message of a count result display box by making the value of the "Data" property into an argument from the carbon button of a linking agency at the time of activation.

[0086] Now, if a setup of a message macro is completed, the "O.K." carbon button 1605 will be clicked, and the setup concerned is made to decide in the message macro setting window of drawing 27 . Then, the contents set up on the message macro setting page with an argument are saved as message macro information as shown in drawing 30 at the item definition file of the OCX concerned, for example, "count result display box .itm."

[0087] A macro name, the class of OCX of a linking agency, a message name, the number of the argument of a message, and the pointer to argument information (after-mentioned) are contained in the message macro information shown in drawing 30 .

[0088] Next, the case where the macro for changing the property which OCX has from the first is set up is explained.

[0089] Drawing 28 is a page which sets up the message macro which changes the property of OCX of the components name for example, of a count result display box.

[0090] If the message macro of a property change is set up, when this OCX will be chosen as a link place at the time of a link setup, this message macro comes to be shown in the message select list of drawing 16 a list table as a message. This function can perform [the processing changed dynamically] the property of OCX now by the design by GUI.

[0091] In drawing 28 , the property change macro [finishing / a setup / already] is shown in the macro name list 1900 the list table. In order to delete the macro whose New button 1901 has been set up to set up a property change macro newly, one macro name to delete from the macro name list 1900 is chosen, and the deletion carbon button 1902 is pushed.

[0092] A push on New button 1901 displays the message macro setting window 2000 of a property change as shown in drawing 29 . A desired macro name is inputted into the macro name input area 2001 in a window 2000. If the carbon button of facing down in the right-hand side of the property name selection field 2002 by which it is indicated by the arrow head is clicked, since a list (property select list) indication of the property which OCX as which it was chosen under current customize has from the first will be given, the property name of OCX to change dynamically is chosen from this property select list, and, subsequently to the modification markup force field 2003, the value after modification of the property concerned is inputted. That is, if the mold of the selected property concerned is character type and it is a numerical mold about an alphabetic character, a numeric value is inputted, and if it is logical type, "TRUE" and "FALSE" are inputted into the modification markup force field 2003.

[0093] If a setup of the message macro of a property change is completed, the "O.K." carbon button 2004 will be clicked, and the setup concerned is made to decide. Then, the contents set up on the message macro setting page of a property change are saved as property change macro information as shown in drawing 31 at the item definition file of the OCX concerned, for example, "count result display box .itm."

[0094] The macro name, the property name, and the value after modification of the property concerned are included in the property change macro information shown in drawing 31 .

[0095] As explained above, an example of item definition FAJ 115 created by operating the customize window of OCX is shown in drawing 32 .

[0096] As shown in drawing 32 , for the item definition file 115 corresponding to one OCX The contents set up on the property Japanese name setting page (property Japanese name information), The contents set up on the event Japanese name setting page (event Japanese name information), The contents set up on the message Japanese name setting page (message Japanese name information), The contents (message macro information) set up on the message macro setting page with an argument and the contents (property change macro information) set up on the message macro setting page of a property change are saved.

[0097] The correspondence relation between the general-purpose property name beforehand given to property Japanese name information to the property which the OCX concerned has, and the Japanese name of the arbitration set up by the user is indicated.

[0098] The correspondence relation between the general-purpose event name which similarly is beforehand given to event Japanese name information to the event which the OCX concerned has, and the Japanese name of the arbitration set up by the user is indicated.

[0099] Moreover, the correspondence relation between the general-purpose message name

beforehand given to message Japanese information to the message which the OCX concerned has, and the Japanese name of the arbitration set up by the user is indicated.

[0100] Message macro information is information as shown in drawing 30 , and indicates the correspondence relation between the message which the OCX concerned has, and OCX of the link origin which can be linked by the desired argument and the message concerned.

[0101] Property change macro information is information as shown in drawing 31 , and indicates correspondence relation with the value to be the property which the OCX concerned has.

[0102] (5) Container ability drawing 5 of a card layout manager (CLM) Index information on OCX that CLM109 is supplied from a server 112 (for the information about the event which the OCX concerned has, a message, a method, and an attribute (property)) Call it hereafter event information, message information, method information, and property information. It is the flow chart which showed the processing actuation which gains property Japanese information, event Japanese information, message Japanese information, message macro information, and property change macro information from the item definition file saved in the database 105.

[0103] If the application of GUI design equipment starts (step S1), CLM109 will acquire OCX information from the registry 113 of a computer 10 (step S2), and as shown in drawing 3 , it will create and arrange the icon of each OCX in a tool bar 204 (step S3).

[0104] The file (icon file) for restoring the icon of OCX is beforehand created corresponding to CLSID, may read the icon file corresponding to CLSID read from registry 113, and may create an icon. Or the icon file of each OCX is memorized by the server 112 corresponding to CLSID of the OCX concerned, is read from a server 112 based on the pointer information to the icon file corresponding to CLSID read from registry 113, and you may make it create an icon.

[0105] Next, based on the OCX information (pointer information included) read from registry 113, CLM109 accesses a server 112, reads the event information which the OCX concerned has about each OCX, message information, property information, and argument information, and creates a table 114 (step S4).

[0106] As the table 114 created here was shown in drawing 6 , one table is assigned for every OCX. That is, one table is made to correspond to CLSID (and components name) of each OCX, and the property information which the OCX has, event information, message information, and argument information are written in it.

[0107] Drawing 7 shows an example of the property information saved on a table 114. CLSID which is the global unique identification information of OCX which has the property concerned in property information, With the member ID which is identification information unique within the OCX concerned for identifying each property when the OCX concerned has two or more properties, if the OCX concerned is a carbon button, for example The property name which shows "data" and the "color of character on screen" showing the graphic character of the carbon button, a text sentence, etc., a "background color", etc., and the mold information on a property that a property shows integer type, character type, etc. are saved. One property (name) supports one property information.

[0108] Since two or more properties usually exist in one OCX, only the number of properties creates this property information, and it is kept on a table 114.

[0109] Drawing 8 shows an example of the event information saved on a table 114. CLSID which is the global unique identification information of OCX which has the event concerned in event information, With the member ID which is identification information unique within the OCX concerned for identifying each event when the OCX concerned has two or more events, if the OCX concerned is a carbon button, for example Event names, such as "a carbon button down", "a carbon button rise", and "a click", and the list of arguments (for example, thing which enumerated parameter identifiers) sent in case the event concerned occurs are saved. One event (name) supports one event information.

[0110] Since two or more events usually exist in one OCX, only the number of events creates this event information, and it is kept on a table 114.

[0111] Drawing 9 shows an example of the message information saved on a table 114. In addition, a message is the function which can call the OCX concerned. The message name of "it being

displayed" as Member ID, "eliminating", etc. which is identification information unique within the OCX concerned for discriminating each message from CLSID which is the global unique identification information of OCX which has the message concerned in message information when the OCX concerned has two or more messages, the list of arguments required when calling a message (parameter identifiers are enumerated), and the return value information on a message are saved. One message (name) corresponds to one message information.

[0112] Since two or more messages usually exist in one OCX, only the number of messages creates this message information, and it is kept on a table 114.

[0113] Drawing 10 shows an example of the argument information referred to from the argument list of event information and message information. To argument information, when Argument ID, parameter identifiers, such as a "figure" and an "alphabetic character color", and a parameter identifier are "figures" as identification information of an argument, the mold of an argument in which integer type or real type is shown is saved.

[0114] Here, CLM109 shall read what is beforehand written in each OCX file, and the property information on a configuration as shown in drawing 7 - drawing 10, event information, message information, and argument information shall create the table 114 as shown in drawing 6.

[0115] Next, CLM109 reads the property Japanese information for every OCX from the item definition file 115 saved in the database 105, and registers the Japanese name of a property into the property information on the table 114 corresponding to the OCX concerned based on it (step S5). That is, a Japanese name is registered into the property information corresponding to the property name included in property Japanese information as shown in drawing 34. In addition, the identification information CLSID of OCX is given to an item definition file, and it can distinguish now of which OCX the item definition file concerned is a thing based on this.

[0116] Moreover, CLM109 reads the event Japanese information for every OCX from the item definition file 115 saved in the database 105, and registers the Japanese name of an event into the event information on the table 114 corresponding to the OCX concerned based on it (step S6). That is, a Japanese name is registered into the event information corresponding to the event name included in event Japanese information as shown in drawing 35.

[0117] Moreover, CLM109 reads the message Japanese information for every OCX from the item definition file 115 saved in the database 105, and registers the Japanese name of a message into the message information on the table 114 corresponding to the OCX concerned based on it (step S7). That is, a Japanese name is registered into the message information corresponding to the message name included in message Japanese information as shown in drawing 36.

[0118] Furthermore, CLM109 reads the message macro information for every OCX, and property change macro information from the item definition file 115 saved in the database 105, and registers them into the table 114 corresponding to the OCX concerned (step S8). Consequently, the table 114 as shown in drawing 33 will be created.

[0119] (6) Explain explanation of design actuation, next design actuation of UI design equipment (inner UI design device 103) of drawing 2. This design actuation is performed by the editing section 107 within UI design device 103 according to the directions from a designer. The condition under design is called design mode. It is necessary to perform two, the screen image (appearance of screen) design which has arranged components which are described below, and the design of a scenario of operation, on UI design equipment of this operation gestalt.

[0120] Here, it explains taking the case of the case where GUI of a calculator as shown in drawing 11 is created.

[0121] (6-1) If the design designer of a screen image starts UI design device 103 in drawing 2, CLM109 in the editing section 107 of the UI design device 103 concerned will be started. the information (screen information [finishing / a design], performance information, material of data, etc.) saved at DB105 if CLM109 is started — memory 106 — reading — ** — as both shown in the flow chart of drawing 5, while CLM109 reads OCX information from registry 113 as it is rare, the item definition file 115 is read from a database 105, and the table 114 as shown in drawing 33 is created.

[0122] As shown in drawing 3, a screen display of the card layout manager menu (CLMM) 200, the card layout manager window (CLMW) 201, and the tool bar 204 is carried out to the display

of an output unit 101. On the other hand, each item of a "card", "a card display", "edit", "an information setup", a "link", and "prototype activation" is prepared for CLMM200.

[0123] As shown in drawing 5, the icon of OCX is displayed on a tool bar 204.

[0124] If a "card" is chosen by a designer operating a data entry unit 102 (for example, mouse) at the time of a design, moving cursor to the item of the "card" in CLMM (card layout manager menu)200 in the state of the screen display of drawing 2, and carrying out a mouse click, the sub menu corresponding to the "card" concerned will be displayed by CLM109. A "new card", "a card components call", and "termination" are prepared for the sub menu of this "card." Then, if a designer makes it move to a "new card" and chooses cursor, a new card will be generated by CLM109 and it will be displayed on CLMW201. The information on this generated card (screen object) is stored in memory 106.

[0125] What is necessary is to display the card 202 of hope and just to arrange the desired components object there, in order to design a screen (UI actuation screen).

[0126] Here, in order to display the card of hope, first, a designer moves cursor to the item of "a card display" in CLMM200, clicks, and displays the sub menu corresponding to the "card display" concerned. The item of "degree card" and a "front card" is prepared for this sub menu. On the other hand, maintenance management is carried out along the order of creation (list logical here) into DB105, and in this phase, the card [finishing / creation] (information) is read into memory 106, as described above.

[0127] Then, the card which it created before one of cards current on display when the card created after one of cards current on display when the designer chose "degree card" chose the "front card" is read from memory 106 by CLM109, and is displayed on CLMW201.

[0128] The card 703 currently displayed on CLMW700 in drawing 11 is the card which did in this way and was read from memory 106, or a card created newly. In addition, although signs that various card components (software components) are arranged are shown by drawing 11 on the card 702, in this phase, it shall not yet be arranged.

[0129] Now, a designer chooses the software components arranged on the card 702 displayed as mentioned above from the icon currently displayed on the tool bar 204 of drawing 3.

[0130] A designer chooses OCX of dedication which performs count called the count result display box 701 from a tool bar 204 first, and specifies the arrangement location of the request on a card 703. Then, CLM109 acquires the pointer information corresponding to the identification information of the selected OCX concerned from registry 113, accesses a server 112, and as an image is created based on the file for restoring the image of the OCX concerned read from the server 112 and it was shown in the specified location on a card 703 at drawing 11, it is arranged.

[0131] Next, a carbon button 702 is chosen from a tool bar 204, and the arrangement location of the request on a card 703 is specified. Then, CLM109 acquires the pointer information corresponding to the identification information of the selected OCX concerned from registry 113, accesses a server 112, and as an image is created based on the file for restoring the image of the OCX concerned read from the server 112 and it was shown in the specified location on a card 703 at drawing 11, it is arranged.

[0132] Next, the property of the software components arranged on a card 703 is set up. For example, the carbon button 702 arranged on a card 703 presupposes that it is OCX with two property names of "data" and an "alphabetic character color" (that is, it has two property information corresponding to the two property names concerned at least). If a designer clicks this carbon button 702 with the mouse as a data entry unit 102 at the time of a design, a property setting dialog box (setting screen) as shown in drawing 12 will be displayed.

[0133] In drawing 12, the mold of the information which can be extracted from the property information on a carbon button 702, i.e., a property name, and a property is a time of this dialog being displayed, and it is desirable to already be displayed on each display box 801, 802, 804, and 805. When the Japanese name explained previously is registered into the property information concerned at this time, it replaces with the official name which the property concerned has from the first, and that Japanese name is displayed on the display box 801.

[0134] A designer should just input a desired text into the set point input edit box 803 of a

property name "data." Supposing it inputs "1 [in this case,]", the data displayed on a carbon button 702 will be set to "1." Moreover, what is necessary is just to input a desired alphabetic character color into the set point input edit box 804 of a property name "an alphabetic character color." Supposing it inputs in this case, "black", the alphabetic character color "1" displayed on a carbon button 702 will become black.

[0135] The above can be repeated and the screen image of a calculator as shown in drawing 11 can be created.

[0136] (6-2) The design designer of a scenario of operation designs a scenario of operation, after ending the design of a screen image. The design of the scenario of operation in this operation gestalt is performed by setting up a link among software components (OCX) for example, by mouse actuation. Here, it carries out by setting up the link which shows "the message of other OCX will be called if an event occurs in a certain OCX." In addition, although there is a link from software components to a card other than the link from software components on software components among the links, in the case of the latter, it mentions later.

[0137] It is describing processing the data (namely, attribute value of a carbon button 702) "1" currently shown in the count result display box 701 by the carbon button 702 if a carbon button's 702 is pushed substituting for setting a link to the count result display box 701 from a carbon button 702 as setting up the link between software components in the case of drawing 11.

[0138] Next, with reference to the flow chart shown in drawing 13, the case where a link is set to the count result display box 701 from a carbon button 702 is taken for an example, and processing actuation of a link setup is explained.

[0139] A designer chooses the carbon button 702 which becomes the beginning a linking agency, and chooses a "link" from the card layout manager menu (CLMM) 200 of drawing 3. Next, the count result display box 701 used as a link place is chosen (step S11). Then, CLM109 displays the link setting dialog 900 as shown in drawing 14 (step S12).

[0140] In drawing 14, the components name "a carbon button 1" of a carbon button 702 and the components name "a count result display box" of the count result display box 701 are already shown in the display edit boxes 902 and 904 of a link place by the link setting dialog 900 the linking agency, respectively. Since the components name is registered into registry 113 corresponding to the identification information CLSID of the OCX concerned, it may read this and may display it on the display edit boxes 902 and 904 of a link place a linking agency.

[0141] A designer inputs a desired link (if designer can grasp, it is good anything) name into the link name setting edit box 901 first. For example, "figure substitution" is inputted here (step S13).

[0142] Next, although the event of OCX which is a linking agency is chosen, the carbon button of facing down in the right-hand side of the event selection field 903 by which it is indicated by the arrow head is clicked in that case. Then, an event select list as shown in drawing 15 is displayed (step S14).

[0143] An event select list reads the event information registered corresponding to the identification information CLSID of a carbon button 702 from the table 114 as shown in drawing 6, and is displayed as a pull down menu which indicated the event name selectable about the OCX concerned which extracted the event name of each event information and was created by the list. Although an event name is the general-purpose name of the actuation in which it may succeed to the OCX concerned, when it is the event name to which the Japanese name is set by the item definition file explained previously, it replaces with the general-purpose name of the event concerned, and the set-up Japanese name is displayed on a pull down menu.

[0144] As the event information on a carbon button 702 is shown in for example, those with four, and drawing 15, each event name is "a carbon button down", "a carbon button rise", "a click", and "MUBU."

[0145] A designer should just choose a desired event name from the event select list of drawing 15 (step S15). Then, the selected event name concerned, for example, "a carbon button down", is displayed on the event selection field 903.

[0146] Next, although the message of OCX which is a link place is chosen, the carbon button of

facing down in the right-hand side of the message selection field 905 by which it is indicated by the arrow head is clicked in that case. Then, a message select list as shown in drawing 16 is displayed (step S16). A message select list is displayed as a pull down menu which indicated the message name selectable about the OCX concerned by the list.

[0147] Here, with reference to drawing 37, message select-list creation / display-processing actuation is explained.

[0148] CLM109 reads the message information registered corresponding to the identification information CLSID of the count result display box 701 which is a link place from the table 114 as shown in drawing 33 (step S101), and creates the pull down menu which indicated the message name selectable about the OCX concerned which extracted the message name of each message information and was created by the list (step S102).

[0149] Furthermore, CLM109 reads message macro information and property change macro information as shown in drawing 30 registered from the table 114 as shown in drawing 33 corresponding to the identification information CLSID of the count result display box 701, and drawing 31 R> 1 (step S103). And when OCX of a linking agency is specified for the message macro information concerned and property change macro information OCX of the link origin previously chosen for the link setup confirms whether be the specified OCX (step S104). When OCX of the link origin previously chosen for the link setup is the link origin specified for the macro information concerned The macro name indicated by the macro information concerned is added to a message select list (step S105), and the message select list concerned is displayed on the link setting dialog of drawing 14 (step S106).

[0150] Thus, the macro name (the macro name of a message macro with an argument, macro name of a property change macro) is also contained in the message name displayed on a message select list besides the general-purpose name of the directions instruction which the OCX concerned which OCX of the link place concerned has can receive. As the message information which the count result display box 701 has from the first is shown in for example, those with five, and drawing 16, each message name is a "figure input", "addition", "subtraction", "multiplication", and a "division", and has "count" as a macro name of a message macro further.

[0151] A designer should just choose a desired message name from the message select list of drawing 16 (step S17). Then, the selected message name concerned, for example, a "figure input", is displayed on the message selection field 905. It judges whether CLM109 has an argument (argument list) from the selected message information concerned stored in the table 114 in that case, and if an argument is needed for the message concerned at the time of calling a link place, the argument setup key 906 will become effective (for example, the argument setup key 906 will be in a display condition from a non-display condition). The dialog for setting up the argument of a message by carrying out the depression of this carbon button is displayed (step S18). After inputting a matter required for this dialog and setting up an argument (step S19), a setup of a link is completed by carrying out the depression of the O.K. carbon button 907 to the last (step S20). That is, the link which has attributes, such as a link place, an event, and a message, is set up link name [which was inputted into the link setting dialog shown in drawing 14], and link origin.

[0152] CLM109 makes a link setting dialog non-display while storing the information on this set-up link in memory 106.

[0153] On the other hand, when the message macro "count" is chosen from the message select list of drawing 16, the selected macro name concerned, i.e., "count", is displayed on the message selection field 905. If this message macro is performed (for example, method information on the macro concerned stored in the database 105 is performed by carrying out the depression of the O.K. carbon button 907), the property "Data" which OCX of link origin called a carbon button has will be substituted for the argument of the name of "Number", and the message passed to OCX of a link place called a count result display box by the message "Insert Number" will be set up automatically. It becomes unnecessary therefore, to carry out a setup of the argument of step S18 - step S19.

[0154] Next, the argument setting procedure which can set drawing 13 step S19 is explained

with reference to the flow chart shown in drawing 17.

[0155] If a designer does the depression of the argument setup key 906 of the link setting dialog of drawing 14, the argument list dialog 1100 as shown in drawing 18 will be displayed (step S31).

[0156] As shown in drawing 18, the message name chosen previously and the argument selection field 1101 which displays that the list of the arguments which the message concerned has is selectable are displayed on the argument list dialog 1100.

[0157] The list of the arguments displayed on the argument selection field 1101 enumerates the parameter identifiers which are what CLM109 created with reference to the argument list contained in the message information on the message chosen previously, for example, are in an argument list.

[0158] Here, since it has the one argument a "figure" from message information, only the argument a "figure" is shown in the message the "figure input" of the count result display box 701 which is OCX of a link place by the argument selection field 1101 of drawing 18. When it, of course, has two or more arguments, all they are displayed on the argument selection field 1101.

[0159] If a designer does the depression of the setup key 1102 further by the argument selection field 1101 of drawing 18 clicking a desired parameter identifier (step S32), the argument setting dialog 1103 as shown in drawing 19 will be displayed (step S33).

[0160] The argument setting dialog 1103 is a dialog which decides what kind of value to set as the argument chosen in the argument list dialog 1100.

[0161] As shown in drawing 19, the parameter identifier which CLM109 read from the argument information concerned on a table 114 to each of the parameter identifier display edit box 1104 and the argument mold display edit box 1105, and the mold of an argument are displayed on the argument setting dialog 1103. Here, it turns out that the value of integer type (int) is set as the argument a "figure."

[0162] In case a designer sets a value as the argument concerned, he chooses whether the value of immobilization is sent as an argument with the set point change radio button 1106, or the property name of OCX is sent as a value of an argument (step S34). In sending a fixed value as an argument, it inputs a desired value (for example, "0") into the set point input edit box 1107 (step S35).

[0163] In sending the property (value) of OCX as an argument, it chooses of which software component (OCX) it is a property name (step S36 - step S37).

[0164] First, the carbon button of facing down in the right-hand side of the components name selection field 1108 by which it is indicated by the arrow head is clicked. Then, a components name select list as shown in drawing 20 is displayed.

[0165] The components name select list shown in drawing 20 is displayed as a pull down menu which indicated by the list all the components names (for example, components name of all OCX arranged on the card 703 in drawing 11) arranged on the card "a linking agency" and under present design by using the property as components available as an argument.

[0166] A designer clicks a desired components name (for example, "linking agency"), and should just choose from the components name select list of drawing 20. Then, the selected components name concerned, for example, "a linking agency", is displayed on the components name selection field 1108.

[0167] Next, the carbon button of facing down in the right-hand side of the property name selection field 1109 by which it is indicated by the arrow head is clicked that the property name which the selected software components (OCX) concerned have should be chosen. Then, a property name select list as shown in drawing 21 is displayed.

[0168] CLM109 reads the property information on the selected software components concerned from a table 114, and the property name select list shown in drawing 21 is displayed as a pull down menu which extracted the property name of the same mold as the mold currently displayed on the argument mold display edit box 1105 among the property names which the selected component concerned has, and indicated by the list. For example, although the property names which this carbon button 702 has are "data" and an "alphabetic character color" since OCX of the link origin concerned is a carbon button 702 when "a linking agency" is chosen from a components name select list, the "data" which is the same mold as the mold currently displayed

on the argument mold display edit box 1105 is displayed on a property name select list.

[0169] A designer clicks a desired property name (for example, "data"), and should just choose from the property name select list of drawing 21. Then, the selected property name concerned, for example, "data", is displayed on the property name selection field 1109.

[0170] A setup of an argument is completed by carrying out the depression of the O.K. carbon button 1101 to the last.

[0171] In the above, although a link setup on software components (OCX) from software components (OCX) was explained, when setting a link to a card from software components, the case where a carbon button 702 and other cards are linked is explained briefly. That is, a designer chooses the carbon button 702 which becomes the beginning a linking agency, and chooses a "link" from the card layout manager menu (CLMM) 200 of drawing 3. Next, the card used as a link place is chosen from the map viewer 205 as shown in drawing 4. Then, CLM109 displays a link setting dialog as shown in drawing 14. In case the event of a linking agency is chosen, it is the same as that of the above-mentioned. In case a different point from the case of the link setting dialog displayed when setting up the link between the software components explained previously chooses the message of the card of a link place, the message select list created without CLM109 referring to a table 114 is displayed. Therefore, also in case an argument required for the message concerned is set up, the setting approach depending on the function of CLM109 can be considered. In addition, since the concrete setting approach is not the summary of this invention, explanation is omitted.

[0172] (6-3) many to many link setting drawing 22 is what showed notionally a link setup in case two or more links origin and two or more link places are chosen, for example, when there is a 2 link place (a components name -- "a carbon button 3", and "a carbon button 4") two links origin (a components name -- "a carbon button 1", and "a carbon button 2"), it can set up a total of four links by actuation of one link setup. In this case, the message and argument which are sent to the event which a linking agency receives, or a link place become common in all links. In addition, it is also the same as when a linking agency or a link place is plurality (one-pair **, many pairs 1).

[0173] Next, only a point which is different from the case of a link setup of the above-mentioned of 1 to 1 with reference to the flow chart of drawing 13 and drawing 17 about such a many to many link configuration procedure is explained.

[0174] In step S11, a designer chooses a "link" from the card layout manager menu (CLMM) 200 of drawing 3, after a click etc. chooses by carrying out two or more software components (for example, a components name "a carbon button 1" and "a carbon button 2") which become a linking agency. Next, a click etc. chooses by carrying out two or more software components (for example, a components name "a carbon button 3" and "a carbon button 4") used as a link place. In the link setting dialog displayed at this time As shown in drawing 23, in the linking agency display edit box 902 as a components name of a linking agency Two or more selected software components names (namely, "a carbon button 1" and "a carbon button 2") concerned are displayed. In the link place display edit box 904 as a components name of a link place Two or more selected software components names (namely, "a carbon button 3" and "a carbon button 4") concerned are displayed.

[0175] When the multiple selection of the linking agency is made, only the event name which the components of two or more selected links origin concerned have in common is displayed on the event select list displayed at step S14. That is, from the table 114 as shown in drawing 6 R> 6, CLM109 extracts only the event name common to two components among the event information registered corresponding to each identification information CLSID of "a carbon button 1" and "a carbon button 2", and creates an event select list.

[0176] When the multiple selection of the link place is made, only the message name which the components of two or more selected link places concerned have in common is displayed on the message select list displayed at step S16. That is, from the table 114 as shown in drawing 6, CLM109 extracts only the message name common to two components among the message information registered corresponding to each identification information CLSID of "a carbon button 3" and "a carbon button 4", and creates a message select list.

[0177] Moreover, when setting up an argument, the property name of software components is made into the value of an argument at step S36 of drawing 17 and "a linking agency" is chosen in the components name selection field 1108 of the argument setting dialog of drawing 19. Only the property name which is the property of the same mold as the mold currently displayed on the argument mold display edit box 1105, and two or more links origin concerned has in common is displayed on the property name select list displayed in case a property name is chosen in the property name selection field 1109. That is, CLM109 is the property of the same mold as the mold currently displayed on the argument mold display edit box 1105 among the property information registered from the table 114 as shown in drawing 6 corresponding to each identification information CLSID of "a carbon button 1" and "a carbon button 2", and extracts only the property name common to two components, and creates a property name select list.

[0178] (7) Explain prototype activation actuation of UI design equipment (inner UI design device 103) of drawing 2 to the explanation last of prototype activation actuation. This prototype activation actuation is performed by the prototype activation section 108 within UI design device 103 according to the directions from a designer. The condition under prototype activation is called prototype execute mode.

[0179] The prototype activation section 108 takes charge of all processings of the data designed in the editing section 107 in prototype execute mode. Although CLM109 in the editing section 107 manages data in design mode as described above (during starting of the editing section 107), PTM111 in the prototype activation section 108 (prototype manager) manages the data recorded on memory 106 in prototype execute mode.

[0180] Now, in order to change from design mode to prototype execute mode, a designer operates a data entry unit 102 (inner mouse) at the time of a design, moves cursor to the item of "prototype activation" in CLMM200 shown in drawing 3, clicks, and should just choose the "prototype activation" concerned. In this case, CLM109 starts PTM111. Thereby, PTM111 displays the prototype manager menu (PTMM) 1201 as shown in drawing 24 on the display of an output unit 101. This PTMM1201 is displayed during starting of PTM111. Each item of "prototype activation", "a prototype halt", and "prototype termination" is prepared for PTMM1201.

[0181] If a designer operates an input unit 104 (inner mouse) in prototype execute mode at the time of simulation and "prototype activation" is chosen from PTMM1201, PTM111 will perform the communication link with the hardware device in an input unit 104 at the time of simulation, and will give an event to the object which corresponds via PTM111 concerned with the data inputted from the hardware device. Thereby, a prototype (simulation) is performed.

[0182] Here, an input of a certain event passes the message set as the link which makes a linking agency the corresponding components to the software components or card of a link place — the software components on a screen are clicked.

[0183] On the other hand, if "a prototype halt" is chosen from PTMM1201, PTM111 will be changed to the condition of not receiving the input from a hardware device, and will interrupt prototype activation temporarily. If "prototype activation" is again chosen in this condition, PTM111 will be canceled of the condition of not receiving the input from a hardware device. Thereby, a prototype is performed again.

[0184] Next, if "prototype termination" is chosen from PTMM1201, termination will be notified to CLM109 from PTM111, and it will become design mode again.

[0185]

[Effect of the Invention] As explained above, according to this invention, the image design of a GUI actuation screen (user interface actuation screen) and a design of operation can carry out easily (not describing a program) by GUI (graphical user interface) actuation using general-purpose software components.

[Translation done.]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-99317

(P 2 0 0 0 - 9 9 3 1 7 A)

(43) 公開日 平成12年4月7日(2000.4.7)

(51) Int. Cl.

G06F 9/06

識別記号

530

F I

G06F 9/06

テーマコード (参考)

530 P 5B076

530 W

審査請求 未請求 請求項の数10 O L (全25頁)

(21) 出願番号 特願平10-265251

(22) 出願日 平成10年9月18日(1998.9.18)

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 雨宮 美香

神奈川県川崎市幸区柳町70番地 株式会社
東芝柳町工場内

(72) 発明者 神尾 広幸

神奈川県川崎市幸区柳町70番地 株式会社
東芝柳町工場内

(74) 代理人 100058479

弁理士 鈴江 武彦 (外6名)

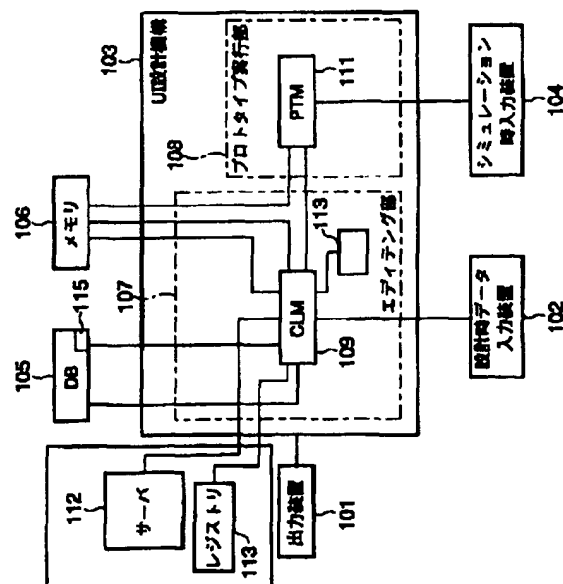
Fターム(参考) 5B076 DC01 DD05 EC06

(54) 【発明の名称】 ユーザインタフェース設計装置及び方法

(57) 【要約】

【課題】 汎用のソフトウェア部品を用いて、GUI 操作画面のイメージ設計と動作の設計がGUI 操作で容易に行うことができる。

【解決手段】 ソフトウェア部品の持つ属性とイベントとメッセージに関するインデックス情報を読み取って、当該読み取られたインデックス情報を記憶手段に記憶し、前記ソフトウェア部品の持つ属性とイベントとメッセージに任意の名称を設定し(アイテム定義ファイル115)、前記記憶手段に記憶されたインデックス情報と前記設定された属性とイベントとメッセージの名称とに基づき、リンク元とリンク先のソフトウェア部品に設定できる属性とイベントとメッセージのリストを作成し、このリストに基づき前記リンク元とリンク先のソフトウェア部品に属性とイベントとメッセージとを設定する。



【特許請求の範囲】

【請求項1】 ユーザインタフェースの操作画面を表すカード上の任意の位置に各種ソフトウェア部品を配置し、任意の前記ソフトウェア部品をリンク元とし、他のソフトウェア部品あるいはカードをリンク先として、該リンク元のイベントと該リンク先のメッセージとを対応付けることで動作の設計を行い、設計したユーザインタフェースのシミュレーションを行うユーザインタフェース設計装置において、

前記ソフトウェア部品の持つ属性とイベントとメッセージに関するインデックス情報を読み取る読取手段と、
前記読取手段で読み取られたインデックス情報を記憶する記憶手段と、

前記ソフトウェア部品の持つ属性とイベントとメッセージに任意の名称を設定する名称設定手段と、

前記記憶手段に記憶されたインデックス情報と前記名称設定手段で設定された属性の名称とに基づき、リンク元とリンク先のソフトウェア部品に設定できる属性のリストを作成する第1の作成手段と、

前記属性リストに基づき前記リンク元とリンク先のソフトウェア部品に属性を設定する属性設定手段と、

前記記憶手段に記憶されたインデックス情報と前記名称設定手段で設定されたイベントの名称とに基づき、リンク元のソフトウェア部品のイベントリストを作成する第2の作成手段と、

前記イベントリストに基づき選択されたイベントを前記リンク元のソフトウェア部品のイベントとして設定するイベント設定手段と、

前記記憶手段に記憶されたインデックス情報と前記名称設定手段で設定されたメッセージの名称とに基づきリンク先のソフトウェア部品のメッセージリストを作成する第3の作成手段と、

前記メッセージリストに基づき選択されたメッセージを前記リンク元のソフトウェア部品に設定されたイベントに対応付ける前記リンク先のソフトウェア部品のメッセージとして設定するメッセージ設定手段と、

を具備したことを特徴とするユーザインタフェース設計装置。

【請求項2】 ユーザインタフェースの操作画面を表すカード上の任意の位置に各種ソフトウェア部品を配置し、任意の前記ソフトウェア部品をリンク元とし、他のソフトウェア部品あるいはカードをリンク先として、該リンク元のイベントと該リンク先のメッセージとを対応付けることで動作の設計を行い、設計したユーザインタフェースのシミュレーションを行うユーザインタフェース設計装置において、

前記ソフトウェア部品の持つ属性とイベントとメッセージに関するインデックス情報を読み取る読取手段と、

前記読取手段で読み取られたインデックス情報を記憶する記憶手段と、

前記ソフトウェア部品の持つメッセージに引数値となるデータを設定して新たなメッセージを作成するメッセージ作成手段と、

前記記憶手段に記憶されたインデックス情報に基づき、リンク元とリンク先のソフトウェア部品に設定できる属性のリストを作成する第1の作成手段と、

前記属性リストに基づき前記リンク元とリンク先のソフトウェア部品に属性を設定する属性設定手段と、

前記記憶手段に記憶されたインデックス情報に基づき、リンク元のソフトウェア部品のイベントリストを作成する第2の作成手段と、

前記イベントリストに基づき選択されたイベントを前記リンク元のソフトウェア部品のイベントとして設定するイベント設定手段と、

前記記憶手段に記憶されたインデックス情報と前記メッセージ作成手段で作成された新たなメッセージとに基づき前記リンク先のソフトウェア部品のメッセージリストを作成する第3の作成手段と、

前記メッセージリストに基づき選択されたメッセージを前記リンク元のソフトウェア部品に設定されたイベントに対応付ける前記リンク先のソフトウェア部品のメッセージとして設定するメッセージ設定手段と、

を具備したことを特徴とするユーザインタフェース設計装置。

【請求項3】 ユーザインタフェースの操作画面を表すカード上の任意の位置に各種ソフトウェア部品を配置し、任意の前記ソフトウェア部品をリンク元とし、他のソフトウェア部品あるいはカードをリンク先として、該リンク元のイベントと該リンク先のメッセージとを対応付けることで動作の設計を行い、設計したユーザインタフェースのシミュレーションを行うユーザインタフェース設計装置において、

前記ソフトウェア部品の持つ属性とイベントとメッセージに関するインデックス情報を読み取る読取手段と、

前記読取手段で読み取られたインデックス情報を記憶する記憶手段と、

前記ソフトウェア部品の持つメッセージに、引数と該メッセージに対応付け可能なリンク元のソフトウェア部品とを設定して、新たなメッセージを作成するメッセージ作成手段と、

前記記憶手段に記憶されたインデックス情報に基づき、リンク元とリンク先のソフトウェア部品に設定できる属性のリストを作成する第1の作成手段と、

前記属性リストに基づき前記リンク元とリンク先のソフトウェア部品に属性を設定する属性設定手段と、

前記記憶手段に記憶されたインデックス情報に基づき、リンク元のソフトウェア部品のイベントリストを作成する第2の作成手段と、

前記イベントリストに基づき選択されたイベントを前記リンク元のソフトウェア部品のイベントとして設定する

イベント設定手段と、

前記記憶手段に記憶されたインデックス情報と前記メッセージ作成手段で作成された新たなメッセージとに基づき、リンク先のソフトウェア部品のメッセージリストを作成する第3の作成手段と、

前記メッセージリストに基づき選択されたメッセージを前記リンク元のソフトウェア部品に設定されたイベントに対応付ける前記リンク先のソフトウェア部品のメッセージとして設定するメッセージ設定手段と、

を具備し、

前記第3の作成手段でメッセージリストを作成する際、前記リンク元のソフトウェア部品が前記新たなメッセージに設定されたリンク元のソフトウェア部品であるときのみ、該新たなメッセージを該メッセージリストに載せることを特徴とするユーザインタフェース設計装置。

【請求項4】 前記メッセージ作成手段で作成される新たなメッセージに設定される引数は、リンク元として設定するソフトウェア部品の属性であることを特徴とする請求項2または3記載のユーザインタフェース設計装置。

【請求項5】 ユーザインタフェースの操作画面を表すカード上の任意の位置に各種ソフトウェア部品を配置し、任意の前記ソフトウェア部品をリンク元とし、他のソフトウェア部品あるいはカードをリンク先として、該リンク元のイベントと該リンク先のメッセージとを対応付けることで動作の設計を行い、設計したユーザインタフェースのシミュレーションを行うユーザインタフェース設計装置において、

前記ソフトウェア部品の持つ属性とイベントとメッセージに関するインデックス情報を読み取る読取手段と、前記読取手段で読み取られたインデックス情報を記憶する記憶手段と、

前記ソフトウェア部品の属性を変更するための新たなメッセージを作成するメッセージ作成手段と、

前記記憶手段に記憶されたインデックス情報に基づき、リンク元とリンク先のソフトウェア部品に設定できる属性のリストを作成する第1の作成手段と、

前記属性リストに基づき前記リンク元とリンク先のソフトウェア部品に属性を設定する属性設定手段と、

前記記憶手段に記憶されたインデックス情報に基づき、リンク元のソフトウェア部品のイベントリストを作成する第2の作成手段と、

前記イベントリストに基づき選択されたイベントを前記リンク元のソフトウェア部品のイベントとして設定するイベント設定手段と、

前記記憶手段に記憶されたインデックス情報と前記メッセージ作成手段で作成されたメッセージとに基づきリンク先のソフトウェア部品のメッセージリストを作成する第3の作成手段と、

前記メッセージリストに基づき選択されたメッセージを

前記リンク元のソフトウェア部品に設定されたイベントに対応付ける前記リンク先のソフトウェア部品のメッセージとして設定するメッセージ設定手段と、

を具備したことを特徴とするユーザインタフェース設計装置。

【請求項6】 ユーザインタフェースの操作画面を表すカード上の任意の位置に各種ソフトウェア部品を配置し、任意の前記ソフトウェア部品をリンク元とし、他のソフトウェア部品あるいはカードをリンク先として、該リンク元のイベントと該リンク先のメッセージとを対応付けることで動作の設計を行い、設計したユーザインタフェースのシミュレーションを行うユーザインタフェース設計方法において、

前記ソフトウェア部品の持つ属性とイベントとメッセージに関するインデックス情報を読み取って、当該読み取られたインデックス情報を記憶手段に記憶し、

前記ソフトウェア部品の持つ属性とイベントとメッセージに任意の名称を設定し、

前記記憶手段に記憶されたインデックス情報と前記設定された属性の名称とに基づき、リンク元とリンク先のソフトウェア部品に設定できる属性のリストを作成し、この属性リストに基づき前記リンク元とリンク先のソフトウェア部品に属性を設定し、

前記記憶手段に記憶されたインデックス情報と前記設定されたイベントの名称とに基づきリンク元のソフトウェア部品のイベントリストを作成し、このイベントリストに基づき選択されたイベントを前記リンク元のソフトウェア部品のイベントとして設定し、

前記記憶手段に記憶されたインデックス情報と前記設定されたメッセージの名称とに基づきリンク先のソフトウェア部品のメッセージリストを作成し、このメッセージリストに基づき選択されたメッセージを前記リンク元のソフトウェア部品に設定されたイベントに対応付ける前記リンク先のソフトウェア部品のメッセージとして設定することを特徴とするユーザインタフェース設計方法。

【請求項7】 ユーザインタフェースの操作画面を表すカード上の任意の位置に各種ソフトウェア部品を配置し、任意の前記ソフトウェア部品をリンク元とし、他のソフトウェア部品あるいはカードをリンク先として、該リンク元のイベントと該リンク先のメッセージとを対応付けることで動作の設計を行い、設計したユーザインタフェースのシミュレーションを行うユーザインタフェース設計方法において、

前記ソフトウェア部品の持つ属性とイベントとメッセージに関するインデック

ス情報を読み取って、当該読み取られたインデックス情報を記憶手段に記憶し、前記ソフトウェア部品の持つメッセージに引数値となるデータを設定して新たなメッセージを作成し、

前記記憶手段に記憶されたインデックス情報に基づき、

10

20

30

40

50

リンク元とリンク先のソフトウェア部品に設定できる属性のリストを作成し、この属性リストに基づき前記リンク元とリンク先のソフトウェア部品に属性を設定し、前記記憶手段に記憶されたインデックス情報に基づきリンク元のソフトウェア部品のイベントリストを作成し、このイベントリストに基づき選択されたイベントを前記リンク元のソフトウェア部品のイベントとして設定し、前記記憶手段に記憶されたインデックス情報と前記作成された新たなメッセージとに基づきリンク先のソフトウェア部品のメッセージリストを作成し、このメッセージ

リストに基づき選択されたメッセージを前記リンク元のソフトウェア部品に設定されたイベントに対応付ける前記リンク先のソフトウェア部品のメッセージとして設定することを特徴とするユーザインタフェース設計方法。

【請求項8】 ユーザインタフェースの操作画面を表すカード上の任意の位置に各種ソフトウェア部品を配置し、任意の前記ソフトウェア部品をリンク元とし、他のソフトウェア部品あるいはカードをリンク先として、該

リンク元のイベントと該リンク先のメッセージとを対応付けることで動作の設計を行い、設計したユーザインタフェースのシミュレーションを行うユーザインタフェース設計方法において、

前記ソフトウェア部品の持つ属性とイベントとメッセージに関するインデックス情報を読み取って、当該読み取られたインデックス情報を記憶手段に記憶し、

前記ソフトウェア部品の持つメッセージに、引数と該メッセージに対応付け可能なリンク元のソフトウェア部品とを設定して、新たなメッセージを作成し、

前記記憶手段に記憶されたインデックス情報に基づき、リンク元とリンク先のソフトウェア部品に設定できる属性のリストを作成し、この属性リストに基づき前記リンク元とリンク先のソフトウェア部品に属性を設定し、前記記憶手段に記憶されたインデックス情報に基づきリンク元のソフトウェア部品のイベントリストを作成し、このイベントリストに基づき選択されたイベントを前記リンク元のソフトウェア部品のイベントとして設定し、前記記憶手段に記憶されたインデックス情報と前記作成された新たなメッセージとに基づきリンク先のソフトウェア部品のメッセージリストを作成し、このメッセージ

リストに基づき選択されたメッセージを前記リンク元のソフトウェア部品に設定されたイベントに対応付ける前記リンク先のソフトウェア部品のメッセージとして設定し、前記メッセージリストを作成する際、前記リンク元のソフトウェア部品が前記新たなメッセージに設定されたリンク元のソフトウェア部品であるときのみ、該新たなメッセージを該メッセージリストに載せることを特徴とするユーザインタフェース設計方法。

【請求項9】 前記新たなメッセージに設定される引数は、リンク元として設定するソフトウェア部品の属性で

あることを特徴とする請求項7または8記載のユーザインタフェース設計方法。

【請求項10】 ユーザインタフェースの操作画面を表すカード上の任意の位置に各種ソフトウェア部品を配置し、任意の前記ソフトウェア部品をリンク元とし、他のソフトウェア部品あるいはカードをリンク先として、該リンク元のイベントと該リンク先のメッセージとを対応付けることで動作の設計を行い、設計したユーザインタフェースのシミュレーションを行うユーザインタフェース設計方法において、

前記ソフトウェア部品の持つ属性とイベントとメッセージに関するインデックス情報を読み取って、当該読み取られたインデックス情報を記憶手段に記憶し、

前記ソフトウェア部品の属性を変更するための新たなメッセージを作成し、

前記記憶手段に記憶されたインデックス情報に基づき、リンク元とリンク先のソフトウェア部品に設定できる属性のリストを作成し、この属性リストに基づき前記リンク元とリンク先のソフトウェア部品に属性を設定し、

前記記憶手段に記憶されたインデックス情報に基づきリンク元のソフトウェア部品のイベントリストを作成し、このイベントリストに基づき選択されたイベントを前記リンク元のソフトウェア部品のイベントとして設定し、前記記憶手段に記憶されたインデックス情報と前記作成された新たなメッセージとに基づきリンク先のソフトウェア部品のメッセージリストを作成し、このメッセージリストに基づき選択されたメッセージを前記リンク元のソフトウェア部品に設定されたイベントに対応付ける前記リンク先のソフトウェア部品のメッセージとして設定することを特徴とするユーザインタフェース設計方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、ユーザからの入力を受け取り、部品をレイアウトした操作画面をユーザインタフェース（UI）として用いたシステムのための画面の設計、動作の設計をグラフィカルユーザインタフェース（GUI）を用いて行い、シミュレーションによる動作確認を行うのに好適なユーザインタフェース設計装置及び方法に関する。

【0002】

【従来の技術】 部品をレイアウトすることによる操作画面設計と、動作設計をGUI操作で実現し、設計通りに実行するツールは数多くある。

【0003】 例えば、Visual Basic（米国Microsoft社の登録商品）を利用すると、GUI操作を利用して簡単にWindows（米国Microsoft社の登録商品）のプログラミングを行うことができる。

【0004】

【発明が解決しようとする課題】 しかしVisual Basic等のビジュアルプログラミングツールは、複雑な操作を行

うユーザインタフェースを構築できる半面、動作設計の大部分はプログラムコードを記述して行うため、設計者がプログラミングを習得する必要がある。

【0005】また、特開平10-21062に記載されているユーザインタフェース設計装置では、装置内で取り扱う部品にあらかじめ属性を設定しており、その属性を選択的に組み合わせてプログラムの記述を行っていた。ただしこの属性にはあらかじめ装置内に保持しておく必要があり、汎用のソフトウェア部品を動的に装置に取り込むことが不可能であった。

【0006】さらに、汎用のソフトウェア部品をある特定の目的を有するユーザインタフェースの設計に用いる場合、以下のような問題点があった。

【0007】(1) 当該ソフトウェア部品の汎用的な名称では、それがどのような部品であるのかわかりにくい。

【0008】(2) 引数を必要とするメッセージをソフトウェア部品間に関連付ける際、予め定められたソフトウェア部品間に同じような引数付きメッセージを関連付ける場合が多く、そのため同じ作業の繰り返し多い。

【0009】(3) 汎用的なソフトウェア部品のプロパティを目的とするユーザインタフェースを設計するために動的に変更できない。

【0010】そこで、本発明は上記事情を考慮してなされたものでその目的は、汎用的なソフトウェア部品を用いた、GUI操作画面(ユーザインタフェース操作画面)のイメージ設計、動作の設計がGUI(グラフィカルユーザインタフェース)操作(イベントとメッセージとを関連付ける操作)で容易に(プログラムを記述する必要なく)行うことができるユーザインタフェース設計装置及び方法を提供することにある。

【0011】また、本発明は、GUI操作画面上に配置する汎用のソフトウェア部品を動的に装置内に組み込むことが可能なユーザインタフェース設計装置および方法を提供することにある。

【0012】

【課題を解決するための手段】請求項1：本発明のユーザインタフェース設計装置は、ユーザインタフェースの操作画面を表すカード上の任意の位置に各種ソフトウェア部品(例えば、OCX)を配置し、任意の前記ソフトウェア部品をリンク元とし、他のソフトウェア部品あるいはカードをリンク先として、該リンク元のイベントと該リンク先のメッセージとを対応付けることで動作の設計を行い、設計したユーザインタフェースのシミュレーションを行うユーザインタフェース設計装置において、前記ソフトウェア部品の持つ属性とイベントとメッセージに関するインデックス情報(プロパティ情報、イベント情報、メッセージ情報、引数情報)を読み取る(レジストリ113からOCX情報を読み取る)読取手段と、前記読取手段で読み取られたインデックス情報を

記憶する記憶手段(テーブル114)と、前記ソフトウェア部品の持つ属性とイベントとメッセージに任意の名称を設定する名称設定手段(アイテム定義ファイル115の作成)と、前記記憶手段に記憶されたインデックス情報と前記名称設定手段で設定された属性の名称(プロパティ日本語名情報)とに基づき、リンク元とリンク先のソフトウェア部品に設定できる属性のリスト(図12参照)を作成する第1の作成手段と、前記属性リストに基づき前記リンク元とリンク先のソフトウェア部品に属性を設定する属性設定手段(リンク設定ダイアログ等)と、前記記憶手段に記憶されたインデックス情報と前記名称設定手段で設定されたイベントの名称(イベント日本語名情報)とに基づき、リンク元のソフトウェア部品のイベントリスト(図15参照)を作成する第2の作成手段と、前記イベントリストに基づき選択されたイベントを前記リンク元のソフトウェア部品のイベントとして設定するイベント設定手段(リンク設定ダイアログ等)と、前記記憶手段に記憶されたインデックス情報と前記名称設定手段で設定されたメッセージの名称(メッセージ日本語名情報)とに基づきリンク先のソフトウェア部品のメッセージリスト(図16参照)を作成する第3の作成手段と、前記メッセージリストに基づき選択されたメッセージを前記リンク元のソフトウェア部品に設定されたイベントに対応付ける前記リンク先のソフトウェア部品のメッセージとして設定するメッセージ設定手段(リンク設定ダイアログ等)と、を具備したことを特徴とする。

【0013】本発明によれば、汎用的なソフトウェア部品を用いて、GUI操作画面(ユーザインタフェース操作画面)のイメージ設計と動作の設計がGUI(グラフィカルユーザインタフェース)操作で容易に(プログラムを記述する必要なく)行うことができる。

【0014】すなわち、GUI操作画面上に配置する汎用のソフトウェア部品を動的に装置内に組み込むことが可能となり、また、これらソフトウェア部品が持つプロパティ、イベント、メッセージのデフォルトの名称ではわかりにくい場合に、GUI設計装置の管理者が設計者の理解しやすい別名をつけることにより、GUI操作画面の設計環境がより向上する。

【0015】請求項2：ユーザインタフェースの操作画面を表すカード上の任意の位置に各種ソフトウェア部品を配置し、任意の前記ソフトウェア部品をリンク元とし、他のソフトウェア部品あるいはカードをリンク先として、該リンク元のイベントと該リンク先のメッセージとを対応付けることで動作の設計を行い、設計したユーザインタフェースのシミュレーションを行うユーザインタフェース設計装置において、前記ソフトウェア部品の持つ属性とイベントとメッセージに関するインデックス情報を読み取る読取手段と、前記読取手段で読み取られたインデックス情報を記憶する記憶手段と、前記ソフト

ウェア部品の持つメッセージに引数値となるデータを設定して新たなメッセージ（引数付きメッセージマクロ）を作成するメッセージ作成手段と、前記記憶手段に記憶されたインデックス情報に基づき、リンク元とリンク先のソフトウェア部品に設定できる属性のリストを作成する第1の作成手段と、前記属性リストに基づき前記リンク元とリンク先のソフトウェア部品に属性を設定する属性設定手段と、前記記憶手段に記憶されたインデックス情報に基づき、リンク元のソフトウェア部品のイベントリストを作成する第2の作成手段と、前記イベントリストに基づき選択されたイベントを前記リンク元のソフトウェア部品のイベントとして設定するイベント設定手段と、前記記憶手段に記憶されたインデックス情報と前記メッセージ作成手段で作成された新たなメッセージとに基づきリンク先のソフトウェア部品のメッセージリストを作成する第3の作成手段と、前記メッセージリストに基づき選択されたメッセージを前記リンク元のソフトウェア部品に設定されたイベントに対応付ける前記リンク先のソフトウェア部品のメッセージとして設定するメッセージ設定手段と、を具備したことを特徴とする。

【0016】本発明によれば、汎用的なソフトウェア部品を用いて、GUI操作画面（ユーザインタフェース操作画面）のイメージ設計と動作の設計がGUI（グラフィカルユーザインタフェース）操作で容易に（プログラムを記述する必要なく）行うことができる。

【0017】すなわち、GUI操作画面上に配置する汎用のソフトウェア部品を動的に装置内に組み込むことが可能となり、また、引数として送るデータの内容が同じメッセージをメッセージマクロとして登録して再利用することができるので、引数を必要とするメッセージを送るリンク設計の手間を省略することを可能にし、GUI操作画面の設計環境がより向上する。

【0018】請求項3： ユーザインタフェースの操作画面を表すカード上の任意の位置に各種ソフトウェア部品を配置し、任意の前記ソフトウェア部品をリンク元とし、他のソフトウェア部品あるいはカードをリンク先として、該リンク元のイベントと該リンク先のメッセージとを対応付けることで動作の設計を行い、設計したユーザインタフェースのシミュレーションを行うユーザインタフェース設計装置において、前記ソフトウェア部品の持つ属性とイベントとメッセージに関するインデックス情報を読み取る読取手段と、前記読取手段で読み取られたインデックス情報を記憶する記憶手段と、前記ソフトウェア部品の持つメッセージに、引数と該メッセージに対応付け可能なリンク元のソフトウェア部品とを設定して、新たなメッセージ（引数付きメッセージマクロ）を作成するメッセージ作成手段と、前記記憶手段に記憶されたインデックス情報に基づき、リンク元とリンク先のソフトウェア部品に設定できる属性のリストを作成する第1の作成手段と、前記属性リストに基づき前記リンク

元とリンク先のソフトウェア部品に属性を設定する属性設定手段と、前記記憶手段に記憶されたインデックス情報に基づき、リンク元のソフトウェア部品のイベントリストを作成する第2の作成手段と、前記イベントリストに基づき選択されたイベントを前記リンク元のソフトウェア部品のイベントとして設定するイベント設定手段と、前記記憶手段に記憶されたインデックス情報と前記メッセージ作成手段で作成された新たなメッセージとに基づき、リンク先のソフトウェア部品のメッセージリストを作成する第3の作成手段と、前記メッセージリストに基づき選択されたメッセージを前記リンク元のソフトウェア部品に設定されたイベントに対応付ける前記リンク先のソフトウェア部品のメッセージとして設定するメッセージ設定手段と、を具備し、前記第3の作成手段でメッセージリストを作成する際、前記リンク元のソフトウェア部品が前記新たなメッセージに設定されたリンク元のソフトウェア部品であるときのみ、該新たなメッセージを該メッセージリストに載せることを特徴とする。

【0019】本発明によれば、汎用的なソフトウェア部品を用いて、GUI操作画面（ユーザインタフェース操作画面）のイメージ設計と動作の設計がGUI（グラフィカルユーザインタフェース）操作で容易に（プログラムを記述する必要なく）行うことができる。

【0020】すなわち、GUI操作画面上に配置する汎用のソフトウェア部品を動的に装置内に組み込むことが可能となり、また、引数として送るデータの内容が同じメッセージ、および、リンク元の属性を引数に送るメッセージの場合に、リンク元を予め限定してメッセージマクロとして登録して再利用することができるので、引数を必要とするメッセージを送るリンク設計の手間を省略でき、GUI操作画面の設計環境がより向上する。

【0021】請求項4： 好ましくは、前記メッセージ作成手段で作成される新たなメッセージに設定される引数は、リンク元として設定するソフトウェア部品の属性であることを特徴とする。

【0022】請求項5： ユーザインタフェースの操作画面を表すカード上の任意の位置に各種ソフトウェア部品を配置し、任意の前記ソフトウェア部品をリンク元とし、他のソフトウェア部品あるいはカードをリンク先として、該リンク元のイベントと該リンク先のメッセージとを対応付けることで動作の設計を行い、設計したユーザインタフェースのシミュレーションを行うユーザインタフェース設計装置において、前記ソフトウェア部品の持つ属性とイベントとメッセージに関するインデックス情報を読み取る読取手段と、前記読取手段で読み取られたインデックス情報を記憶する記憶手段と、前記ソフトウェア部品の属性を変更するための新たなメッセージを作成するメッセージ作成手段と、前記記憶手段に記憶されたインデックス情報に基づき、リンク元とリンク先のソフトウェア部品に設定できる属性のリストを作成する

第1の作成手段と、前記属性リストに基づき前記リンク元とリンク先のソフトウェア部品に属性を設定する属性設定手段と、前記記憶手段に記憶されたインデックス情報に基づき、リンク元のソフトウェア部品のイベントリストを作成する第2の作成手段と、前記イベントリストに基づき選択されたイベントを前記リンク元のソフトウェア部品のイベントとして設定するイベント設定手段と、前記記憶手段に記憶されたインデックス情報と前記メッセージ作成手段で作成されたメッセージとに基づきリンク先のソフトウェア部品のメッセージリストを作成する第3の作成手段と、前記メッセージリストに基づき選択されたメッセージを前記リンク元のソフトウェア部品に設定されたイベントに対応付ける前記リンク先のソフトウェア部品のメッセージとして設定するメッセージ設定手段と、を具備したことを特徴とする。

【0023】本発明によれば、汎用的なソフトウェア部品を用いて、GUI操作画面（ユーザインタフェース操作画面）のイメージ設計と動作の設計がGUI（グラフィカルユーザインタフェース）操作で容易に（プログラムを記述する必要なく）行うことができる。

【0024】すなわち、GUI操作画面上に配置する汎用のソフトウェア部品を動的に装置内に組み込むことが可能となり、また、GUI操作画面の設計時に、ソフトウェア部品のプロパティを動的に変更するためのメッセージを用いることができ、GUI操作画面の設計環境がより向上する。

【0025】請求項6：本発明のユーザインタフェース設計方法は、ユーザインタフェースの操作画面を表すカード上の任意の位置に各種ソフトウェア部品を配置し、任意の前記ソフトウェア部品をリンク元とし、他のソフトウェア部品あるいはカードをリンク先として、該リンク元のイベントと該リンク先のメッセージとを対応付けることで動作の設計を行い、設計したユーザインタフェースのシミュレーションを行うユーザインタフェース設計方法において、前記ソフトウェア部品の持つ属性とイベントとメッセージに関するインデックス情報を読み取って、当該読み取られたインデックス情報を記憶手段に記憶し、前記ソフトウェア部品の持つ属性とイベントとメッセージに任意の名称を設定し、前記記憶手段に記憶されたインデックス情報と前記設定された属性の名称とに基づき、リンク元とリンク先のソフトウェア部品に設定できる属性のリストを作成し、この属性リストに基づき前記リンク元とリンク先のソフトウェア部品に属性を設定し、前記記憶手段に記憶されたインデックス情報と前記設定されたイベントの名称とに基づきリンク先のソフトウェア部品のイベントリストを作成し、このイベントリストに基づき選択されたイベントを前記リンク元のソフトウェア部品のイベントとして設定し、前記記憶手段に記憶されたインデックス情報と前記設定されたメッ

ッセージリストを作成し、このメッセージリストに基づき選択されたメッセージを前記リンク元のソフトウェア部品に設定されたイベントに対応付ける前記リンク先のソフトウェア部品のメッセージとして設定することを特徴とする。

【0026】発明によれば、汎用的なソフトウェア部品を用いて、GUI操作画面（ユーザインタフェース操作画面）のイメージ設計と動作の設計がGUI（グラフィカルユーザインタフェース）操作で容易に（プログラムを記述する必要なく）行うことができる。

【0027】すなわち、GUI操作画面上に配置する汎用のソフトウェア部品を動的に装置内に組み込むことが可能となり、また、これらソフトウェア部品が持つプロパティ、イベント、メッセージのデフォルトの名称ではわかりにくい場合に、GUI設計装置の管理者が設計者の理解しやすい別名をつけることにより、GUI操作画面の設計環境がより向上する。

【0028】請求項7：本発明のユーザインタフェース設計方法は、ユーザインタフェースの操作画面を表すカード上の任意の位置に各種ソフトウェア部品を配置し、任意の前記ソフトウェア部品をリンク元とし、他のソフトウェア部品あるいはカードをリンク先として、該リンク元のイベントと該リンク先のメッセージとを対応付けることで動作の設計を行い、設計したユーザインタフェースのシミュレーションを行うユーザインタフェース設計方法において、前記ソフトウェア部品の持つ属性とイベントとメッセージに関するインデックス情報を読み取って、当該読み取られたインデックス情報を記憶手段に記憶し、前記ソフトウェア部品の持つメッセージに引数値となるデータを設定して新たなメッセージを作成し、前記記憶手段に記憶されたインデックス情報に基づき、リンク元とリンク先のソフトウェア部品に設定できる属性のリストを作成し、この属性リストに基づき前記リンク元とリンク先のソフトウェア部品に属性を設定し、前記記憶手段に記憶されたインデックス情報に基づきリンク元のソフトウェア部品のイベントリストを作成し、このイベントリストに基づき選択されたイベントを前記リンク元のソフトウェア部品のイベントとして設定し、前記記憶手段に記憶されたインデックス情報と前記作成された新たなメッセージとに基づきリンク先のソフトウェア部品のメッセージリストを作成し、このメッセージリストに基づき選択されたメッセージを前記リンク元のソフトウェア部品に設定されたイベントに対応付ける前記リンク先のソフトウェア部品のメッセージとして設定することを特徴とする。

【0029】本発明によれば、汎用的なソフトウェア部品を用いて、GUI操作画面（ユーザインタフェース操作画面）のイメージ設計と動作の設計がGUI（グラフィカルユーザインタフェース）操作で容易に（プログラムを記述する必要なく）行うことができる。

【0030】すなわち、GUI 操作画面上に配置する汎用のソフトウェア部品を動的に装置内に組み込むことが可能となり、また、引数として送るデータの内容が同じメッセージをメッセージマクロとして登録して再利用することができるので、引数を必要とするメッセージを送るリンク設計の手間を省略することを可能にし、GUI 操作画面の設計環境がより向上する。

【0031】請求項 8： 本発明のユーザインタフェース設計方法は、ユーザインタフェースの操作画面を表すカード上の任意の位置に各種ソフトウェア部品を配置し、任意の前記ソフトウェア部品をリンク元とし、他のソフトウェア部品あるいはカードをリンク先として、該リンク元のイベントと該リンク先のメッセージとを対応付けることで動作の設計を行い、設計したユーザインタフェースのシミュレーションを行うユーザインタフェース設計方法において、前記ソフトウェア部品の持つ属性とイベントとメッセージに関するインデックス情報を読み取って、当該読み取られたインデックス情報を記憶手段に記憶し、前記ソフトウェア部品の持つメッセージに、引数と該メッセージに対応付け可能なリンク元のソフトウェア部品とを設定して、新たなメッセージを作成し、前記記憶手段に記憶されたインデックス情報に基づき、リンク元とリンク先のソフトウェア部品に設定できる属性のリストを作成し、この属性リストに基づき前記リンク元とリンク先のソフトウェア部品に属性を設定し、前記記憶手段に記憶されたインデックス情報に基づきリンク元のソフトウェア部品のイベントリストを作成し、このイベントリストに基づき選択されたイベントを前記リンク元のソフトウェア部品のイベントとして設定し、前記記憶手段に記憶されたインデックス情報と前記作成された新たなメッセージとに基づきリンク先のソフトウェア部品のメッセージリストを作成し、このメッセージリストに基づき選択されたメッセージを前記リンク元のソフトウェア部品に設定されたイベントに対応付ける前記リンク先のソフトウェア部品のメッセージとして設定し、前記メッセージリストを作成する際、前記リンク元のソフトウェア部品が前記新たなメッセージに設定されたリンク元のソフトウェア部品であるときのみ、該新たなメッセージを該メッセージリストに載せることを特徴とする。

【0032】本発明によれば、汎用的なソフトウェア部品を用いて、GUI 操作画面（ユーザインタフェース操作画面）のイメージ設計と動作の設計が GUI（グラフィカルユーザインタフェース）操作で容易に（プログラムを記述する必要なく）行うことができる。

【0033】すなわち、GUI 操作画面上に配置する汎用のソフトウェア部品を動的に装置内に組み込むことが可能となり、また、引数として送るデータの内容が同じメッセージ、および、リンク元の属性を引数に送るメッセージの場合に、リンク元を予め限定してメッセージマ

クロとして登録して再利用することができるので、引数を必要とするメッセージを送るリンク設計の手間を省略でき、GUI 操作画面の設計環境がより向上する。

【0034】請求項 9： 好ましくは、前記新たなメッセージに設定される引数は、リンク元として設定するソフトウェア部品の属性であることを特徴とする。

【0035】請求項 10： 本発明のユーザインタフェース設計方法は、ユーザインタフェースの操作画面を表すカード上の任意の位置に各種ソフトウェア部品を配置し、任意の前記ソフトウェア部品をリンク元とし、他のソフトウェア部品あるいはカードをリンク先として、該リンク元のイベントと該リンク先のメッセージとを対応付けることで動作の設計を行い、設計したユーザインタフェースのシミュレーションを行うユーザインタフェース設計方法において、前記ソフトウェア部品の持つ属性とイベントとメッセージに関するインデックス情報を読み取って、当該読み取られたインデックス情報を記憶手段に記憶し、前記ソフトウェア部品の属性を変更するための新たなメッセージを作成し、前記記憶手段に記憶されたインデックス情報に基づき、リンク元とリンク先のソフトウェア部品に設定できる属性のリストを作成し、この属性リストに基づき前記リンク元とリンク先のソフトウェア部品に属性を設定し、前記記憶手段に記憶されたインデックス情報に基づきリンク元のソフトウェア部品のイベントリストを作成し、このイベントリストに基づき選択されたイベントを前記リンク元のソフトウェア部品のイベントとして設定し、前記記憶手段に記憶されたインデックス情報と前記作成された新たなメッセージとに基づきリンク先のソフトウェア部品のメッセージリストを作成し、このメッセージリストに基づき選択されたメッセージを前記リンク元のソフトウェア部品に設定されたイベントに対応付ける前記リンク先のソフトウェア部品のメッセージとして設定することを特徴とする。

【0036】本発明によれば、汎用的なソフトウェア部品を用いて、GUI 操作画面（ユーザインタフェース操作画面）のイメージ設計と動作の設計が GUI（グラフィカルユーザインタフェース）操作で容易に（プログラムを記述する必要なく）行うことができる。

【0037】すなわち、GUI 操作画面上に配置する汎用のソフトウェア部品を動的に装置内に組み込むことが可能となり、また、GUI 操作画面の設計時に、ソフトウェア部品のプロパティを動的に変更するためのメッセージを用いることができ、GUI 操作画面の設計環境がより向上する。

【0038】

【発明の実施の形態】以下、本発明の実施の形態について図面を参照して説明する。

【0039】（1） ユーザインタフェース設計装置の構成

まず、ユーザインタフェース設計装置の構成について説明する。

【0040】本実施形態にかかるユーザインタフェース（UI）設計装置は、図2に示すように、主に、音声や音（サウンド）の出力に用いられるスピーカ、及びCRTディスプレイ、液晶ディスプレイ等の表示装置を含む出力装置101と、キーボードやマウス等の入力手段からなり、設計時のデータ等の入力に用いられる入力装置（以下、設計時データ入力装置と称する）102と、設計時及びシミュレーション時に設計者あるいは利用者とのインタラクティブな対話を行うユーザインタフェース設計機構（以下、UI設計機構と称する）103と、

（タッチ入力用の）タッチパネル、（音声入力用の）マイクロフォン、マウス、キーボード等の入力手段からなり、シミュレーション時の情報入力に用いられる入力装置（以下、シミュレーション時入力装置と称する）104と、設計された画面情報、動作情報及びデータの素材等を保存・管理するデータベース（DB）105と、設計された画面情報、動作情報及びデータの素材等を一時保持するための領域、作業領域等を提供するメモリ106とで構成されている。ここで、設計時データ入力装置102及びシミュレーション時入力装置104は、物理的には同一の入力装置である。データベース105には、後述するアイテム定義ファイル115が格納される。

【0041】本実施形態にかかるユーザインタフェース設計装置は、例えば、タッチパネル、マイクロフォン、マウス、キーボード等の入力手段、スピーカ、ディスプレイ（表示装置）、等の出力手段、フロッピーディスク装置、ハードディスク装置等の大容量記憶装置、主メモリ、及びCPU等を備えたパーソナルコンピュータ（計算機）を用いて実現される。

【0042】特に、図2のユーザインタフェース設計装置の中心をなすUI設計機構103の持つ機能は、その機能を上記のパーソナルコンピュータにより実行させるためのプログラムを記録した記憶媒体、例えばフロッピーディスクを用い、当該記録媒体を上記パーソナルコンピュータ中のフロッピーディスク装置に装着して、当該フロッピーディスクに記録されているプログラムを上記パーソナルコンピュータ（中のCPU）で読み取り実行させることにより実現される。

【0043】例えば、図1に示すように、記録媒体に記録されたUI設計機構103の持つ機能を実行させるためのプログラムはコンピュータ10上で実行され、コンピュータ10には入力手段としてのキーボード11およびマウス12、出力手段としてのディスプレイ13が接続されている。

【0044】UI設計機構103は、画面や動作の設計時の管理を司るエディティング部107と、設計したUIの動作確認のためのシミュレーション時の管理を司る

プロトタイプ実行部108とで構成されている。このエディティング部107及びプロトタイプ実行部108の構成については後述する。

【0045】（2）UI設計機構103の概要

次に、UI設計機構103の概要について、図3の画面イメージ例を参照して説明する。

【0046】図2のUI設計機構103は、UIの画面使用、操作仕様をGUIで設計するツールである。UI設計機構103による画面の設計は、図3に示すように、出力装置101としてのディスプレイ13に表示される画面上で行われる。すなわち、カードレイアウトマネージャウインドウ（CLMW）201上でUI操作画面を表す画面オブジェクト（オブジェクト指向でいうところのオブジェクト）としてのカード202を作成し、当該カード上にソフトウェア部品202を配置することで画面設計が行われる。

【0047】ソフトウェア部品は、ツールバー204から任意のものが選択されて、配置される。本実施形態では、このソフトウェア部品として、例えば、Win32（米国Microsoft社マイクロソフト社の登録商品）API（Application Program Interface）上で用いられるOCX（OLE（Object Linking and Embedding）カスタムコントロール）を用いる。

【0048】各OCXには、それぞれCLSIDと呼ばれるソフトウェア部品の種類を表したグローバルユニークな識別情報を有する。また、OCXのファイル（実体ファイル）には、CLSIDの他に、後述するように、当該OCXが持つ属性、イベント、メッセージ等のインデックス情報（イベント情報、メッセージ情報、プロパティ情報、引数情報）や、メソッド等が記述されている。

【0049】OCXを供給するものが、図2のサーバ112である。ユーザが当該GUI設計装置で利用したいOCXは、予め（UI設計装置を起動して設計を開始する前に）、コンピュータ10にインストールされている必要がある。すなわち、図1のコンピュータ10にハードディスク等の記憶装置にフロッピーディスクやCD-ROMなどを使用してOCXをインストールすると、コンピュータ10の所定の記憶領域に当該OCXのファイルが書き込まれる。OCXのファイルが書き込まれた記憶領域をサーバ112と呼ぶ。その際、OS（オペレーティングシステム）に当該インストールされたOCXのCLSIDを登録する必要があるが、ここでは、例えば、Windows（米国Microsoft社の登録商品）のレジストリが用いられる。

【0050】OCXがインストールされると、レジストリ113には、当該OCXのCLSIDと、それに対応して、当該OCXの名前（例えば、「ボタン」、「計算結果表示箱」というような部品名）やOCXのファイル

の記憶されている記憶領域へのポインタ情報等（以下、OCX情報と呼ぶ）が登録される。

【0051】このようなOCXのインストール（サーバ112の作成、レジストリ113への登録）は、この場合、Windowsに対応したインストーラにより自動的に実行される。

【0052】なお、UI設計機構103による設計時には、CLMW201と共にマップビューア205が表示されてもよい。このマップビューア205は、作成済みのカードを一覧して表示するためのウインドウであり、当該マップビューア205上には、作成済みのカードの縮小イメージであるカードアイコン206が表示される。作成済みのカード数が多い場合には、一部のカードに対応するカードアイコンのみがマップビューア205上に表示され、残りのカードに対応するカードアイコンは当該マップビューア205上で上下或いは左右にスクロールすることで表示される。また、カード上に配置されるソフトウェア部品（以下、部品オブジェクトとも呼ぶ）から他のカードへのリンクが張られている場合には、対応するカードアイコン間にリンクとその方向を表す矢印（リンク線）が表示される。

【0053】（3） マネジャの種類

次に、図2中のエディティング部107及びプロトタイプ実行部108が有するマネジャ（管理手段）の種類について説明する。

【0054】まず、図2中のエディティング部107は、カード202の生成及び管理と、ソフトウェア部品をサーバ112から読み込み、カード202上に配置するカードレイアウトマネジャ（CLM）109から構成される。

【0055】CLM109は、起動時には、カード202及びカード202上にレイアウトされたソフトウェア部品の編集、及びユーザからのイベント（ボタンをクリックする、音声による入力を行うなど）に対して行われる動作処理のためのリンク設定を行う。また、ソフトウェア部品（すなわち、OCX）をサーバ112から読み出し、カード202に登録する、いわゆるコンテナとしての機能を有する。

【0056】一方、図2中のプロトタイプ実行部108は、プロトタイプマネジャ（PTM）111により構成される。PTM111は、ユーザからの指示に従ってCLM109により起動される。このPTM111の起動時には、ユーザからのイベントを受けたソフトウェア部品は、そのメッセージ送信機能により、設定されたリンク先へメッセージを送る。このメッセージ送信の繰り返しにより、リンクの設計通りに動作シミュレーションを行うことができる。

【0057】（4） アイテム定義ファイルの作成

図2のユーザインタフェース設計装置を用いてユーザインタフェースを設計する際に、より使い易い環境を提供

するため、当該ユーザインタフェース設計装置は、汎用的なソフトウェア部品であるOCXをカスタマイズするための機能（ソフトウェア部品に対する任意の名称の付加機能、引数付きメッセージのマクロの設定機能、プロパティ変更マクロの設定機能）を有している。

【0058】図25は、CLM109により出力装置101に表示されるOCXのカスタマイズウインドウの一例を示したものである。図25では、例えば、部品名が「ボタン」であるOCXのカスタマイズウインドウを示している。例えば、出力装置101に表示されている図3に示すような表示画面上にあるツールバー204から所望のOCXを選択することにより表示される、当該OCXに対する操作メニューからカスタマイズを指示することにより、図25に示すようなウインドウが表示される。

【0059】図25に示すように、OCXのカスタマイズウインドウには、5ページ分のタブ1301a～1301eが表示され、各タブ1301a～1301eに対応するページ1302上で、それぞれOCXのプロパティ（属性）の日本語名、イベントの日本語名、メッセージの日本語名、引数付きメッセージのマクロ、プロパティ変更のマクロの設定を行うようになっている。

【0060】このカスタマイズウインドウで設定した内容は、アイテム定義ファイル115として、各OCX毎にデータベース105に例えばテキスト形式で保存される。アイテム定義ファイル115は、1つのOCXにつき、「当該OCXの部品名、.itm」というファイル名を持つものとする。

【0061】まず、当該選択されたOCXがもつプロパティ、イベント、メッセージにユーザが所望する任意の日本語名を設定する場合について説明する。

【0062】図25では、部品名が「ボタン」のOCXに対し、そのOCXが持つプロパティにユーザが所望する任意の日本語名を設定するためのプロパティ日本語名設定ページ1302が選択されている。

【0063】リストボックス1303の「プロパティ名」というフィールドには、当該選択されているOCXの予めもつプロパティ名の一覧が縦一列に表示される。このプロパティ名のうち所望のものをクリックして選択すると、テキスト入力ボックス1304が表示される。

【0064】テキスト入力ボックス1304に所望の日本語名を入力して、当該入力を確定すると、テキスト入力ボックス1304が消え、当該選択されたプロパティ名の右隣のフィールドに、入力した日本語名が表示される。

【0065】イベント、メッセージの日本語名もそれぞれのページを選択、表示して上記同様の手順にて設定すればよい。

【0066】プロパティ、イベント、メッセージの日本語名設定ページ1302上での設定が終了したら、「O

K」ボタン1305をクリックして、当該設定を確定させる。すると、これらページ上で設定されたプロパティ、イベント、メッセージの日本語名は、当該OCXのアイテム定義ファイル、例えば、「ボタン. itm」に保存される。

【0067】次に、引数付きメッセージマクロ（以下、簡単にメッセージマクロと呼ぶことがある）を設定する場合について説明する。

【0068】図26は、例えば、計算結果表示箱という部品名のOCXの引数つきメッセージのマクロを設定するページである。

【0069】引数つきメッセージのメッセージマクロを設定しておく、後述するリンク設定時に、このOCXがリンク先に選択されると、メッセージとしてこのマクロが図16のメッセージ選択リストにリスト表示されるようになり、既に引数が設定されているメッセージとして（リンク設定時には引数を設定しなくても）利用することができる。この機能により、同じ内容の引数でメッセージを送る場合には、引数の設定を省略した形でリンク設定を行えるようになるので設計者の手間を省くことができる。

【0070】図26において、マクロ名リスト1500には、既に設定済みのメッセージマクロのマクロ名がリスト表示されている。新しくメッセージマクロを設定するには新規作成ボタン1501を、既設定のマクロを削除するには、マクロ名リスト1500から削除したいマクロ名を1つ選択して削除ボタン1502を押す。

【0071】新規作成ボタン1501を押すと、図27に示すような引数メッセージマクロ設定ウインドウ1600が表示される。ここで、マクロ名入力領域1601に、所望のマクロ名を入力し、リンク元選択領域1602に、リンク元となるOCXの種類（部品名）を選択して入力し、メッセージ選択領域1603にメッセージを選択して入力する。リンク元選択領域1602、メッセージ選択領域1603でリンク元やメッセージを選択する際には、当該選択されたOCXが予め持つ選択可能なリンク元やメッセージの羅列したリンク元選択リストやメッセージ選択リストが表示されて、そのリストから所望のものを選択するようにしてしてもよい。このとき、リンク元選択リストからリンク元「なし」を選択することもできる。引数にリンク元のプロパティを設定する必要がない場合や、リンク設定時にすべてのリンク元からこのメッセージマクロを呼べるようにしたい（リンク元を限定したくない）場合には、リンク元「なし」を選択する。それ以外の場合はリンク元のOCXの種類を選択する。

【0072】その後、引数ボタン1604を押すと、例えば、図18に示すような引数一覧ダイアログ1100が表示される。

【0073】図18に示したように、引数一覧ダイア

ログ1100には、先に選択されたメッセージ名と、当該メッセージの有する引数の一覧を選択可能なように表示する引数選択領域1101とが表示されている。

【0074】引数選択領域1101に表示される引数の一覧は、CLM109が、先に選択されたメッセージのメッセージ情報に含まれる引数リストを参照して作成したもので、例えば、引数リストにある引数名を羅列したものである。

【0075】ここで、リンク先のOCXである計算結果表示箱701の「数字入力」というメッセージには、メッセージ情報から「数字」という引数を1つ有している。図18の引数選択領域1101には「数字」という引数のみが表示されている。もちろん引数を複数有する場合には、それら全てが引数選択領域1101に表示される。

【0076】図18の引数選択領域1101から所望の引数名をクリックして選択し、さらに、設定ボタン1102を押下すると、図19に示したような引数設定ダイアログ1103が表示される。引数設定ダイアログ1103は、当該選択されたメッセージにどんな値の引数を設定するかを決めるダイアログである。

【0077】図19に示したように、引数設定ダイアログ1103には、引数名表示エディットボックス1104、引数型表示エディットボックス1105のそれぞれに、CLM109がテーブル114の当該引数情報から読み出した引数名と引数の型とが表示されている。ここでは、「数字」という引数に整数型（int）の値が設定されることが分かる。

【0078】当該引数に値を設定する際、設定値切替えラジオボタン1106で固定の値を引数として送るか、OCXのプロパティ名を引数の値として送るかを選択する。固定値を引数として送る場合には、設定値入力エディットボックス1107に所望の値（例えば「0」）を入力する。

【0079】OCXのプロパティ（の値）を引数として送る場合には、どのソフトウェア部品（OCX）のプロパティ名であるかを選択する。まず、部品名選択領域1108の右側にある下向きの矢印表示されているボタンをクリックする。すると、部品名選択リストが表示されるが、ここでは、「リンク元」「リンク先」の2つのみをしか選択することができない（なお、図27でリンク元を「なし」が選ばれているときは「リンク先」のみ選択可能である）。これは、メッセージマクロ設定時には、リンク設定時に画面上にどのようなOCXが作成されているかわからず、わかっている情報としてはリンク設定時に必ず作成されているOCXはリンク元とリンク先に指定されたOCXの2種類のみ（リンク元とリンク先のOCXが同じ場合は1種類）であるという事実のみだからである。

【0080】部品名選択リストで「リンク元」を選び、

次に、プロパティ名選択領域 1109 の右側にある下向きの矢印表示されているボタンをクリックする。すると、プロパティ名選択リストが表示される。プロパティ名選択リストには、図 27 のリンク元選択領域 1601 で選択した OCX のプロパティ名がリスト表示される。部品名選択リストで「リンク先」を選択したときに表示されるプロパティ選択リストには、現在選択中の（カスタマイズ中である）OCX のプロパティ名がリスト表示される。このプロパティ名選択リストから所望のプロパティ名をクリックして選択すればよい。

【0081】なお、プロパティ名選択リストは、CLM 1109 が部品名選択リストから選択した部品のプロパティ情報をテーブル 114 から読み出して、当該選択された部品の有するプロパティ名のうち、引数型表示エディットボックス 1105 に表示されている型と同じ型のプロパティ名を抜き出して一覧表示したプルダウンメニューとして表示される。

【0082】最後に OK ボタン 1101 を押下することで、引数の設定を完了する。

【0083】ここで、計算結果表示箱という OCX の「Number」という名の引数を 1 つ持つメッセージ「Insert Number」のメッセージマクロを作成する場合を例にとり、より具体的に説明する。図 27 のメッセージマクロ設定ウインドウ 1600 のマクロ名入力領域 1601 に、「計算」というマクロ名を入力し、リンク元選択領域 1602 で、リンク元として「ボタン」という OCX を選択し、メッセージ選択領域 1603 で、「Insert Number」というメッセージを選択する。

【0084】さらに、引数ボタン 1604 を押下して、引数一覧ダイアログ 1100 で引数名「Number」を選択し、ついで、引数設定ダイアログ 1103 では、設定値切替えラジオボタン 1106 で引数にプロパティを送るように設定する。すなわち、部品名選択領域 1108 で「リンク元」を選択すると、プロパティ名選択領域 1109 には、リンク元選択領域 1602 でリンク元として選択した「ボタン」という OCX のプロパティがプロパティ名選択リストが表示されるので、そこから「Data」というプロパティを選択し、「OK」ボタン 1100 を押す。このように作成した計算結果表示箱という OCX の「計算」メッセージマクロは、ボタンという OCX から計算結果表示箱という OCX へのリンク設定を行うときにメッセージとして利用できるようになる。

【0085】「計算」メッセージマクロを用いて設定したリンクは、実行時にリンク元のボタンから「Data」プロパティの値を引数として計算結果表示箱の「Insert Number」メッセージを呼ぶコードに変換される。

【0086】さて、図 27 のメッセージマクロ設定ウインドウにて、メッセージマクロの設定が終了したら「O

K」ボタン 1605 をクリックして、当該設定を確定させる。すると、引数付きメッセージマクロ設定ページ上で設定された内容が、図 30 に示すようなメッセージマクロ情報として、当該 OCX のアイテム定義ファイル、例えば、「計算結果表示箱. itm」に保存される。

【0087】図 30 に示すメッセージマクロ情報には、例えば、マクロ名、リンク元の OCX の種類、メッセージ名、メッセージの引数の個数、引数情報（後述）へのポインタが含まれている。

10 【0088】次に、OCX がもともと持っているプロパティを変更するためのマクロを設定する場合について説明する。

【0089】図 28 は、例えば、計算結果表示箱という部品名の OCX のプロパティを変更するメッセージマクロを設定するページである。

【0090】プロパティ変更のメッセージマクロを設定しておく、リンク設定時にこの OCX がリンク先に選択されると、メッセージとしてこのメッセージマクロが図 16 のメッセージ選択リストにリスト表示されるようになる。この機能により、OCX のプロパティを動的に変更する処理が GUI による設計で行えるようになる。

【0091】図 28 において、マクロ名リスト 1900 には、既に設定済みのプロパティ変更マクロがリスト表示されている。新しくプロパティ変更マクロを設定するには新規作成ボタン 1901 を、既設定のマクロを削除するには、マクロ名リスト 1900 から削除したいマクロ名を 1 つ選択して削除ボタン 1902 を押す。

【0092】新規作成ボタン 1901 を押すと、図 29 に示すようなプロパティ変更のメッセージマクロ設定ウインドウ 2000 が表示される。ウインドウ 2000 中のマクロ名入力領域 2001 に、所望のマクロ名を入力する。プロパティ名選択領域 2002 の右側にある下向きの矢印表示されているボタンをクリックすると、現在カスタマイズ中の選択された OCX がもともと持っているプロパティのリスト（プロパティ選択リスト）表示されるので、このプロパティ選択リストから動的に変更したい OCX のプロパティ名を選択し、ついで、変更値入力領域 2003 に当該プロパティの変更後の値を入力する。すなわち、当該選択したプロパティの型が文字型なら文字を、数値型なら数値を入力し、論理型なら「TRUE」か「FALSE」を変更値入力領域 2003 に入力する。

【0093】プロパティ変更のメッセージマクロの設定が終了したら「OK」ボタン 2004 をクリックして、当該設定を確定させる。すると、プロパティ変更のメッセージマクロ設定ページ上で設定された内容が、図 31 に示すようなプロパティ変更マクロ情報として、当該 OCX のアイテム定義ファイル、例えば、「計算結果表示箱. itm」に保存される。

50 【0094】図 31 に示すプロパティ変更マクロ情報に

は、例えば、マクロ名、プロパティ名、当該プロパティの変更後の値が含まれている。

【0095】以上説明したように、OCXのカスタマイズウィンドウを操作することにより作成された、アイテム定義ファイル115の一例を図32に示す。

【0096】図32に示すように、1つのOCXに対応するアイテム定義ファイル115には、プロパティ日本語名設定ページで設定された内容（プロパティ日本語名情報）と、イベント日本語名設定ページで設定された内容（イベント日本語名情報）と、メッセージ日本語名設定ページで設定された内容（メッセージ日本語名情報）と、引数付きメッセージマクロ設定ページで設定された内容（メッセージマクロ情報）と、プロパティ変更のメッセージマクロ設定ページで設定された内容（プロパティ変更マクロ情報）とが保存されている。

【0097】プロパティ日本語名情報には、当該OCXが有するプロパティに対し予め与えられている汎用的なプロパティ名と、ユーザにより設定された任意の日本語名との対応関係が記載されている。

【0098】同様に、イベント日本語名情報には、当該OCXが有するイベントに対し予め与えられている汎用的なイベント名と、ユーザにより設定された任意の日本語名との対応関係を記載したものである。

【0099】また、メッセージ日本語名情報には、当該OCXが有するメッセージに対し予め与えられている汎用的なメッセージ名と、ユーザにより設定された任意の日本語名との対応関係を記載したものである。

【0100】メッセージマクロ情報は、例えば、図30に示したような情報で、当該OCXが有するメッセージと、所望の引数、当該メッセージにてリンクできるリンク元のOCXとの対応関係を記載したものである。

【0101】プロパティ変更マクロ情報は、例えば、図31に示したような情報で、当該OCXが有するプロパティと、その値との対応関係を記載したものである。

【0102】(5) カードレイアウトマネージャ (CLM) のコンテナ機能

図5は、CLM109がサーバ112から供給されるOCXのインデックス情報（当該OCXの持つイベント、メッセージ、メソッド、属性（プロパティ）についての情報で、以下、イベント情報、メッセージ情報、メソッド情報、プロパティ情報と呼ぶ）、データベース105に保存されているアイテム定義ファイルからプロパティ日本語名情報、イベント日本語名情報、メッセージ日本語名情報、メッセージマクロ情報、プロパティ変更マクロ情報を得る処理動作を示したフローチャートである。

【0103】GUI設計装置のアプリケーションが起動すると（ステップS1）、CLM109は、コンピュータ10のレジストリ113からOCX情報を取得し（ステップS2）、図3に示したように、ツールバー204に各OCXのアイコンを作成し並べる（ステップS

3）。

【0104】OCXのアイコンを復元するためのファイル（アイコンファイル）は、例えば、CLSIDに対応して予め作成されているものであって、レジストリ113から読み出したCLSIDに対応したアイコンファイルを読み出してアイコンを作成してもよい。あるいは、各OCXのアイコンファイルは、サーバ112に当該OCXのCLSIDに対応して記憶され、レジストリ113から読み出したCLSIDに対応したアイコンファイルへのポインタ情報に基づきサーバ112から読み出してアイコンを作成するようにしてもよい。

【0105】次に、CLM109は、レジストリ113から読み出したOCX情報（に含まれるポインタ情報）に基づき、サーバ112にアクセスし、各OCXについて、当該OCXが持つイベント情報、メッセージ情報、プロパティ情報、引数情報を読み出して、テーブル114を作成する（ステップS4）。

【0106】ここで作成されるテーブル114は、例えば、図6に示したように、各OCX毎に1つのテーブルが割り当てられている。すなわち、1つのテーブルには、各OCXのCLSID（および部品名）に対応させて、そのOCXが有するプロパティ情報、イベント情報、メッセージ情報、引数情報が書き込まれている。

【0107】図7は、テーブル114に保存されるプロパティ情報の一例を示したものである。プロパティ情報には、当該プロパティを持つOCXのグローバルユニークな識別情報であるCLSIDと、当該OCXが複数のプロパティを有する場合に各プロパティを識別するための当該OCX内でユニークな識別情報であるメンバIDと、例えば、当該OCXがボタンであるならば、そのボタンの表示文字、テキスト文等を表す「データ」や「表示文字色」や「背景色」等を示すプロパティ名と、プロパティが整数型か文字型か等を示すプロパティの型情報とが保存されている。1つのプロパティ情報には、1つのプロパティ（名）が対応している。

【0108】1つのOCXには通常複数のプロパティが存在するので、このプロパティ情報をプロパティの数だけ作成し、テーブル114に保管する。

【0109】図8は、テーブル114に保存されるイベント情報の一例を示したものである。イベント情報には、当該イベントを持つOCXのグローバルユニークな識別情報であるCLSIDと、当該OCXが複数のイベントを有する場合に各イベントを識別するための当該OCX内でユニークな識別情報であるメンバIDと、例えば、当該OCXがボタンであるならば、「ボタンダウン」「ボタンアップ」「クリック」等のイベント名と、当該イベントが発生する際に送られる引数のリスト（例えば、引数名を羅列したもの）とが保存されている。1つのイベント情報には1つのイベント（名）が対応して

【0110】1つのOCXには通常複数のイベントが存在するので、このイベント情報をイベントの数だけ作成し、テーブル114に保管する。

【0111】図9は、テーブル114に保存されるメッセージ情報の一例を示したものである。なお、メッセージとは当該OCXを呼び出すことが可能な関数のことである。メッセージ情報には、当該メッセージを持つOCXのグローバルユニークな識別情報であるCLSIDと、当該OCXが複数のメッセージを有する場合には各メッセージを識別するための当該OCX内でユニークな識別情報であるメンバIDと、「表示する」「消去する」等のメッセージ名と、メッセージを呼び出す時に必要な引数のリスト（例えば、引数名を羅列）と、メッセージの戻り値情報とが保存される。1つのメッセージ情報には1つのメッセージ（名）が対応する。

【0112】1つのOCXには通常複数のメッセージが存在するので、このメッセージ情報をメッセージの数だけ作成し、テーブル114に保管する。

【0113】図10は、イベント情報、メッセージ情報の引数リストから参照される引数情報の一例を示す。引数情報には、引数の識別情報として引数IDと、「数字」「文字色」等の引数名と、例えば、引数名が「数字」のとき整数型か実数型かを示す引数の型とが保存されている。

【0114】ここでは、図7～図10に示したような構成のプロパティ情報、イベント情報、メッセージ情報、引数情報は、予め各OCXファイルに書き込まれているものをCLM109が読み出して、図6に示したようなテーブル114を作成するものとする。

【0115】次に、CLM109は、データベース105に保存されているアイテム定義ファイル115から各OCX毎のプロパティ日本語情報を読み込み、それを基に、プロパティの日本語名を当該OCXに対応するテーブル114のプロパティ情報に登録する（ステップS5）。すなわち、プロパティ日本語情報に含まれるプロパティ名に対応するプロパティ情報に、図34に示すように日本語名を登録する。なお、アイテム定義ファイルにはOCXの識別情報CLSIDが付され、これに基づき当該アイテム定義ファイルはどのOCXのものであるかが判別できるようになっていてもよい。

【0116】また、CLM109は、データベース105に保存されているアイテム定義ファイル115から各OCX毎のイベント日本語情報を読み込み、それを基に、イベントの日本語名を当該OCXに対応するテーブル114のイベント情報に登録する（ステップS6）。すなわち、イベント日本語情報に含まれるイベント名に対応するイベント情報に、図35に示すように日本語名を登録する。

【0117】また、CLM109は、データベース105に保存されているアイテム定義ファイル115から各

OCX毎のメッセージ日本語情報を読み込み、それを基に、メッセージの日本語名を当該OCXに対応するテーブル114のメッセージ情報に登録する（ステップS7）。すなわち、メッセージ日本語情報に含まれるメッセージ名に対応するメッセージ情報に、図36に示すように日本語名を登録する。

【0118】さらに、CLM109は、データベース105に保存されているアイテム定義ファイル115から各OCX毎のメッセージマクロ情報、プロパティ変更マクロ情報を読み込み、それらを、当該OCXに対応するテーブル114に登録する（ステップS8）。その結果、図33に示すようなテーブル114が作成されることになる。

【0119】(6) 設計操作の説明

次に、図2のUI設計装置（中のUI設計機構103）の設計操作について説明する。この設計操作は、設計者からの指示に従ってUI設計機構103内のエディティング部107によって実行される。設計中の状態を、設計モードという。本実施形態のUI設計装置上では、以下に述べるような、部品を配置した画面イメージ（画面の外観）設計と、動作シナリオの設計の2つを行う必要がある。

【0120】ここでは、図11に示すような電卓のGUIを作成する場合を例にとり説明する。

【0121】(6-1) 画面イメージの設計

設計者が図2中のUI設計機構103を立ち上げると、当該UI設計機構103のエディティング部107内のCLM109が起動される。CLM109が起動されると、DB105に保存されている情報（設計済みの画面情報、動作情報及びデータの素材等）がメモリ106に読み込まれと共に、図5のフローチャートに示したように、CLM109は、レジストリ113からOCX情報を読み込むとともに、データベース105からアイテム定義ファイル115を読み込み、図33に示したようなテーブル114を作成する。

【0122】図3に示されるように、カードレイアウトマネージャメニュー（CLMM）200とカードレイアウトマネージャウインドウ（CLMW）201とツールバー204が出力装置101のディスプレイに画面表示される。一方、CLMM200には、例えば、「カード」「カード表示」「編集」「情報設定」「リンク」「プロトタイプ実行」の各項目が用意されている。

【0123】ツールバー204には、図5に示したようにOCXのアイコンが表示される。

【0124】図2の画面表示状態で、設計者が設計時データ入力装置102（の例えばマウス）を操作し、CLMM（カードレイアウトマネージャメニュー）200中の「カード」の項目にカーソルを移動させてマウスクリックすることで「カード」を選択すると、当該「カード」に対応するサブメニューがCLM109により表示され

る。この「カード」のサブメニューには、例えば「新規カード」「カード部品呼び出し」「終了」が用意されている。そこで、設計者がカーソルを「新規カード」へ移動させて選択すると、CLM109により新規のカードが生成され、CLMW201上に表示される。この生成されたカードの情報（画面オブジェクト）は、メモリ106に格納される。

【0125】画面（UI操作画面）を設計するには、希望のカード202を表示させ、そこに所望の部品オブジェクトを配置していけばよい。

【0126】ここで、希望のカードを表示させるには、まず設計者がCLMM200中の「カード表示」の項目にカーソルを移動させクリックし、当該「カード表示」に対応するサブメニューを表示させる。このサブメニューには、例えば「次カード」と「前カード」という項目が用意されている。一方、作成済みのカード（の情報）はDB105内に作成順の並び（ここでは論理的な並び）で保持管理されており、この段階では前記したようにメモリ106に読み込まれている。

【0127】そこで、設計者が「次カード」を選択すると現在表示中のカードの1枚後に作成したカードが、「前カード」を選択すると現在表示中のカードの1枚前に作成したカードが、CLM109によりメモリ106から読み込まれ、CLMW201に表示される。

【0128】図11においてCLMW700に表示されているカード703は、このようにしてメモリ106から読み込まれたカード、或いは新規に作成されたカードである。なお、図11では、カード702上に各種カード部品（ソフトウェア部品）が配置されている様子が示されているが、この段階では未だ配置されていないものとする。

【0129】さて設計者は、上記のようにして表示されたカード702に配置するソフトウェア部品を、図3のツールバー204に表示されているアイコンから選択する。

【0130】設計者は、まず、ツールバー204から計算結果表示箱701という計算を行う専用のOCXを選択し、カード703上の所望の配置位置を指定する。すると、CLM109は、レジストリ113から当該選択されたOCXの識別情報に対応するポインタ情報を取得してサーバ112にアクセスし、サーバ112から読み出された当該OCXのイメージを復元するためのファイルに基づきイメージを作成してカード703上の指定位置に図11に示したように配置する。

【0131】次に、ツールバー204からボタン702を選択して、カード703上の所望の配置位置を指定する。すると、CLM109は、レジストリ113から当該選択されたOCXの識別情報に対応するポインタ情報を取得してサーバ112にアクセスし、サーバ112から読み出された当該OCXのイメージを復元するための

ファイルに基づきイメージを作成してカード703上の指定位置に図11に示したように配置する。

【0132】次に、カード703上に配置されたソフトウェア部品のプロパティを設定する。例えば、カード703上に配置されたボタン702は、「データ」、「文字色」という2つのプロパティ名を持つ（すなわち、少なくとも当該2つのプロパティ名に対応する2つのプロパティ情報を有する）OCXであるとする。設計者は、このボタン702を設計時データ入力装置102としてのマウスでクリックすると、図12に示したようなプロパティ設定ダイアログボックス（設定画面）が表示される。

【0133】図12において、ボタン702のプロパティ情報から抽出できる情報、すなわち、プロパティ名とプロパティの型は、このダイアログが表示される時点ですでにそれぞれの表示ボックス801、802、804、805に表示されていることが望ましい。このとき、当該プロパティ情報に先に説明した日本語名が登録されている場合には、表示ボックス801に、当該プロパティがもともと持っている正式名称に代えて、その日本語名が表示される。

【0134】設計者は、プロパティ名「データ」の設定値入力エディットボックス803に所望のテキストを入力すればよい。この場合、例えば、「1」を入力したとすると、ボタン702に表示されるデータは「1」となる。また、プロパティ名「文字色」の設定値入力エディットボックス804に所望の文字色を入力すればよい。この場合、例えば、「黒」を入力したとすると、ボタン702に表示される「1」という文字色は黒色となる。

【0135】以上を繰り返して、図11に示したような電卓の画面イメージを作成することができる。

【0136】（6-2） 動作シナリオの設計

設計者は、画面イメージの設計を終了すると、動作シナリオの設計を行う。本実施形態における動作シナリオの設計は、例えばマウス操作によってソフトウェア部品（OCX）間にリンクを設定することで行われる。ここでは、「あるOCXにイベントが発生すると他のOCXのメッセージを呼び出す」ということを示すリンクを設定することで行う。なお、リンクにはソフトウェア部品からソフトウェア部品へのリンクの他にソフトウェア部品からカードへのリンクがあるが、後者の場合は後述する。

【0137】ソフトウェア部品間のリンクを設定するとは、例えば、図11の場合、ボタン702から計算結果表示箱701へリンクを設定することで、ボタン702が押されたら計算結果表示箱701にボタン702に表示されているデータ（すなわち、ボタン702の属性値）「1」が代入するという処理を記述することである。

【0138】次に、図13に示すフローチャートを参照

して、ボタン702から計算結果表示箱701へリンクを設定する場合を例にとり、リンク設定の処理動作について説明する。

【0139】設計者は最初にリンク元となるボタン702を選択し、図3のカードレイアウトマネージャメニュー（CLMM）200から「リンク」を選択する。次にリンク先となる計算結果表示箱701を選択する（ステップS11）。するとCLM109は図14に示したようなリンク設定ダイアログ900を表示する（ステップS12）。

【0140】図14において、リンク設定ダイアログ900には、リンク元とリンク先の表示エディットボックス902、904には、それぞれ、ボタン702の部品名「ボタン1」と計算結果表示箱701の部品名「計算結果表示箱」とが既に表示されている。部品名は、レジストリ113に当該OCXの識別情報CLSIDに対応して登録されているので、これを読み取って、リンク元とリンク先の表示エディットボックス902、904に表示してもよい。

【0141】設計者は、まず、リンク名設定エディットボックス901に所望の（設計者が把握することのできるものであれば何でもよい）リンク名を入力する。例えば、ここでは、「数字代入」が入力されている（ステップS13）。

【0142】次に、リンク元であるOCXのイベントを選択するわけであるが、その際、イベント選択領域903の右側にある下向きの矢印表示されているボタンをクリックする。すると、図15に示したようなイベント選択リストが表示される（ステップS14）。

【0143】イベント選択リストは、図6に示したようなテーブル114からボタン702の識別情報CLSIDに対応して登録されているイベント情報を読み込み、各イベント情報のイベント名を抜き出して作成された当該OCXについて選択可能なイベント名を一覧表示したプルダウンメニューとして表示される。イベント名は、当該OCXに対し為され得る操作の汎用的な名称であるが、先に説明したアイテム定義ファイルにて日本語名が設定されているイベント名の場合は、当該イベントの汎用的な名称に代えて、その設定された日本語名がプルダウンメニューに表示される。

【0144】ボタン702のイベント情報は例えば4つあり、図15に示すように、それぞれのイベント名は「ボタンダウン」「ボタンアップ」「クリック」「ムーブ」である。

【0145】設計者は、図15のイベント選択リストから所望のイベント名を選択すればよい（ステップS15）。すると、イベント選択領域903には、当該選択されたイベント名、例えば、「ボタンダウン」が表示される。

【0146】次に、リンク先であるOCXのメッセージ

を選択するわけであるが、その際、メッセージ選択領域905の右側にある下向きの矢印表示されているボタンをクリックする。すると、図16に示したようなメッセージ選択リストが表示される（ステップS16）。メッセージ選択リストは、当該OCXについて選択可能なメッセージ名を一覧表示したプルダウンメニューとして表示される。

【0147】ここで、図37を参照して、メッセージ選択リスト作成/表示処理動作について説明する。

【0148】CLM109は、図33に示したようなテーブル114からリンク先である計算結果表示箱701の識別情報CLSIDに対応して登録されているメッセージ情報を読み込み（ステップS101）、各メッセージ情報のメッセージ名を抜き出して作成された当該OCXについて選択可能なメッセージ名を一覧表示したプルダウンメニューを作成する（ステップS102）。

【0149】さらに、CLM109は、図33に示したようなテーブル114から計算結果表示箱701の識別情報CLSIDに対応して登録されている図30、図31に示すようなメッセージマクロ情報、プロパティ変更マクロ情報を読み込む（ステップS103）。そして、当該メッセージマクロ情報およびプロパティ変更マクロ情報にて、リンク元のOCXが指定されているときは、リンク設定のために先に選択されたリンク元のOCXが、その指定されたOCXであるか否かチェックして（ステップS104）、リンク設定のために先に選択したリンク元のOCXが当該マクロ情報にて指定されたリンク元であるときは、当該マクロ情報に記載されているマクロ名をメッセージ選択リストに追加し（ステップS105）、図14のリンク設定ダイアログ上に当該メッセージ選択リストを表示する（ステップS106）。

【0150】このように、メッセージ選択リストに表示されるメッセージ名には、当該リンク先のOCXの有する当該OCXが受け付け得る指示命令の汎用的な名称の他に、マクロ名（引数付きメッセージマクロのマクロ名、プロパティ変更マクロのマクロ名）も含まれている。計算結果表示箱701がもともと持つメッセージ情報は例えば5つあり、図16に示すように、それぞれのメッセージ名は「数字入力」「加算」「減算」「乗算」「除算」であり、さらに、メッセージマクロのマクロ名として「計算」がある。

【0151】設計者は、図16のメッセージ選択リストから所望のメッセージ名を選択すればよい（ステップS17）。すると、メッセージ選択領域905には、当該選択されたメッセージ名、例えば、「数字入力」が表示される。その際、CLM109は、テーブル114に格納された当該選択されたメッセージ情報から引数（引数リスト）があるか否かを判断し、リンク先を呼び出す際の当該メッセージに引数を必要とするものであれば、引数設定ボタン906が有効となる（例えば、引数設定が

タン906は、非表示状態から表示状態となる)。このボタンを押下することでメッセージの引数を設定するためのダイアログが表示される(ステップS18)。このダイアログに必要な事項を入力して、引数の設定を行った後(ステップS19)、最後にOKボタン907を押下することで、リンクの設定を完了する(ステップS20)。すなわち、図14に示したリンク設定ダイアログに入力されたリンク名、リンク元、リンク先、イベント、メッセージ等の属性を有するリンクが設定される。

【0152】CLM109は、この設定されたリンクの情報をメモリ106に格納すると共にリンク設定ダイアログを非表示とする。

【0153】一方、図16のメッセージ選択リストから「計算」というメッセージマクロを選択した場合、メッセージ選択領域905には、当該選択されたマクロ名、すなわち、「計算」が表示される。このメッセージマクロが実行されれば(例えば、OKボタン907を押下することで、データベース105に格納されている当該マクロのメソッド情報を実行する)、ボタンというリンク元のOCXのもつ「Data」というプロパティを「Number」という名の引数に代入して、「Insert Number」というメッセージにて計算結果表示箱というリンク先のOCXへ渡すメッセージを自動的に設定する。従って、ステップS18～ステップS19の引数の設定はする必要がなくなる。

【0154】次に、図13のステップS19における引数設定処理手順について、図17に示すフローチャートを参照して説明する。

【0155】設計者が図14のリンク設定ダイアログの引数設定ボタン906を押下すると、図18に示したような引数一覧ダイアログ1100が表示される(ステップS31)。

【0156】図18に示したように、引数一覧ダイアログ1100には、先に選択されたメッセージ名と、当該メッセージの有する引数の一覧を選択可能なように表示する引数選択領域1101とが表示されている。

【0157】引数選択領域1101に表示される引数の一覧は、CLM109が、先に選択されたメッセージのメッセージ情報に含まれる引数リストを参照して作成したもので、例えば、引数リストにある引数名を羅列したものである。

【0158】ここで、リンク先のOCXである計算結果表示箱701の「数字入力」というメッセージには、メッセージ情報から「数字」という引数を1つ有している。図18の引数選択領域1101には「数字」という引数のみが表示されている。もちろん引数を複数有する場合には、それら全てが引数選択領域1101に表示される。

【0159】設計者は、図18の引数選択領域1101が所望の引数名をクリックし、さらに、設定ボタン11

02を押下すると(ステップS32)、図19に示したような引数設定ダイアログ1103が表示される(ステップS33)。

【0160】引数設定ダイアログ1103は、引数一覧ダイアログ1100で選択した引数にどんな値を設定するかを決めるダイアログである。

【0161】図19に示したように、引数設定ダイアログ1103には、引数名表示エディットボックス1104、引数型表示エディットボックス1105のそれぞれに、CLM109がテーブル114の当該引数情報から読み出した引数名と引数の型とが表示されている。ここでは、「数字」という引数に整数型(int)の値が設定されることが分かる。

【0162】設計者は、当該引数に値を設定する際、設定値切替えラジオボタン1106で固定の値を引数として送るか、OCXのプロパティ名を引数の値として送るかを選択する(ステップS34)。固定値を引数として送る場合には、設定値入力エディットボックス1107に所望の値(例えば「0」)を入力する(ステップS35)。

【0163】OCXのプロパティ(の値)を引数として送る場合には、どのソフトウェア部品(OCX)のプロパティ名であるかを選択する(ステップS36～ステップS37)。

【0164】まず、部品名選択領域1108の右側にある下向きの矢印表示されているボタンをクリックすると、図20に示したような部品名選択リストが表示される。

【0165】図20に示す部品名選択リストは、そのプロパティを引数として利用可能な部品として、「リンク元」と、現在設計中のカード上に配置されている全ての部品名(例えば、図11においてカード703上に配置されている全てのOCXの部品名)を一覧表示したブルダウンメニューとして表示される。

【0166】設計者は、図20の部品名選択リストから所望の部品名(例えば、「リンク元」)をクリックして選択すればよい。すると、部品名選択領域1108には、当該選択された部品名、例えば、「リンク元」が表示される。

【0167】次に、当該選択されたソフトウェア部品(OCX)の持つプロパティ名を選択すべく、プロパティ名選択領域1109の右側にある下向きの矢印表示されているボタンをクリックする。すると、図21に示したようなプロパティ名選択リストが表示される。

【0168】図21に示すプロパティ名選択リストは、CLM109が当該選択されたソフトウェア部品のプロパティ情報をテーブル114から読み出して、当該選択された部品の有するプロパティ名のうち、引数型表示エディットボックス1105に表示されている型と同じ型のプロパティ名を抜き出して一覧表示したブルダウンメ

ニューとして表示される。例えば、部品名選択リストから「リンク元」を選択した場合、当該リンク元のOCXはボタン702であるので、このボタン702の有するプロパティ名は「データ」と「文字色」であるが、そのうち、引数型表示エディットボックス1105に表示されている型と同じ型である「データ」のみがプロパティ名選択リストに表示される。

【0169】設計者は、図21のプロパティ名選択リストから所望のプロパティ名（例えば、「データ」）をクリックして選択すればよい。すると、プロパティ名選択領域1109には、当該選択されたプロパティ名、例えば、「データ」が表示される。

【0170】最後にOKボタン1101を押下することで、引数の設定を完了する。

【0171】以上、ソフトウェア部品（OCX）からソフトウェア部品（OCX）へのリンク設定について説明したが、ソフトウェア部品からカードへリンクを設定する場合、例えば、ボタン702と他のカードをリンクする場合について簡単に説明する。すなわち、設計者は最初にリンク元となるボタン702を選択し、図3のカードレイアウトマネージャメニュー（CLMM）200から「リンク」を選択する。次にリンク先となるカードを、たとえば、図4に示したようなマップビューア205から選択する。するとCLM109は図14に示したようなリンク設定ダイアログを表示する。リンク元のイベントを選択する際は、前述同様である。先に説明したソフトウェア部品間のリンクを設定する際に表示されたリンク設定ダイアログの場合と異なる点は、リンク先のカードのメッセージを選択する際は、CLM109がテーブル114を参照することなく作成したメッセージ選択リストが表示される。従って、当該メッセージに必要な引数を設定する際にも、CLM109の機能に依存する設定方法が考えられる。なお、その具体的な設定方法は、本発明の要旨ではないので説明は割愛する。

【0172】（6-3）多対多のリンク設定

図22は、複数のリンク元や複数のリンク先が選択されている場合のリンク設定を概念的に示したもので、例えば、2つのリンク元（部品名が「ボタン1」と「ボタン2」）と2つリンク先（部品名が「ボタン3」と「ボタン4」）がある場合、1回のリンク設定の操作で合計4つのリンクが設定できる。この場合、リンク元が受けるイベントやリンク先に送るメッセージ、引数はすべてのリンクにおいて共通となる。なお、リンク元とリンク先のいずれか一方が複数である場合（1対多、多対1）も同様である。

【0173】次に、このような多対多のリンク設定手順について、図13、図17のフローチャートを参照して、前述の1対1のリンク設定の場合と異なる点のみ説明する。

【0174】ステップS11において、設計者はリンク

元となる複数のソフトウェア部品（例えば、部品名「ボタン1」と「ボタン2」）をクリック等して選択した後、図3のカードレイアウトマネージャメニュー（CLMM）200から「リンク」を選択する。次にリンク先となる複数のソフトウェア部品（例えば、部品名「ボタン3」と「ボタン4」）をクリック等して選択する。このとき表示されるリンク設定ダイアログには、図23に示すように、リンク元表示エディットボックス902にはリンク元の部品名として、当該選択された複数のソフトウェア部品名（すなわち、「ボタン1」と「ボタン2」）が表示され、リンク先表示エディットボックス904にはリンク先の部品名として、当該選択された複数のソフトウェア部品名（すなわち、「ボタン3」と「ボタン4」）が表示されている。

【0175】ステップS14で表示されるイベント選択リストには、リンク元が複数選択されているとき、当該選択された複数のリンク元の部品が共通に有するイベント名のみが表示される。すなわち、CLM109は、図6に示したようなテーブル114から、「ボタン1」「ボタン2」のそれぞれの識別情報CLSIDに対応して登録されているイベント情報のうち2つの部品に共通しているイベント名のみを抜き出してイベント選択リストを作成する。

【0176】ステップS16で表示されるメッセージ選択リストには、リンク先が複数選択されているとき、当該選択された複数のリンク先の部品が共通に有するメッセージ名のみが表示される。すなわち、CLM109は、図6に示したようなテーブル114から、「ボタン3」「ボタン4」のそれぞれの識別情報CLSIDに対応して登録されているメッセージ情報のうち2つの部品に共通しているメッセージ名のみを抜き出してメッセージ選択リストを作成する。

【0177】また、引数を設定する際、図17のステップS36でソフトウェア部品のプロパティ名を引数の値とする場合に、図19の引数設定ダイアログの部品名選択領域1108で「リンク元」を選択したとき、プロパティ名選択領域1109でプロパティ名を選択する際に表示されるプロパティ名選択リストには、引数型表示エディットボックス1105に表示されている型と同じ型のプロパティで、かつ、当該複数のリンク元が共通にもつプロパティ名のみが表示される。すなわち、CLM109は、図6に示したようなテーブル114から「ボタン1」「ボタン2」のそれぞれの識別情報CLSIDに対応して登録されているプロパティ情報のうち、引数型表示エディットボックス1105に表示されている型と同じ型のプロパティで、かつ、2つの部品に共通しているプロパティ名のみを抜き出してプロパティ名選択リストを作成する。

【0178】（7）プロトタイプ実行操作の説明
最後に、図2のUI設計装置（中のUI設計機構10

3) のプロトタイプ実行操作について説明する。このプロトタイプ実行操作は、設計者からの指示に従ってUI設計機構103内のプロトタイプ実行部108によって実行される。プロトタイプ実行中の状態を、プロトタイプ実行モードという。

【0179】プロトタイプ実行モードでは、エディティング部107で設計したデータの処理は、全てプロトタイプ実行部108が担当する。設計モードでは(エディティング部107の起動中は)、前記したようにエディティング部107内のCLM109がデータの管理を行うが、プロトタイプ実行モードでは、プロトタイプ実行部108内のPTM(プロトタイプマネージャ)111がメモリ106に記録されたデータの管理を行う。

【0180】さて、設計モードからプロトタイプ実行モードに切り替えるには、設計者は設計時データ入力装置102(中のマウス)を操作して、図3に示したCLM200中の「プロトタイプ実行」の項目にカーソルを移動させてクリックし、当該「プロトタイプ実行」を選択すればよい。この場合、CLM109はPTM111を起動する。これによりPTM111は、図24に示すようなプロトタイプマネージャメニュー(PTMM)1201を出力装置101のディスプレイに表示する。このPTMM1201は、PTM111の起動中表示される。PTMM1201には、「プロトタイプ実行」「プロトタイプ一時停止」「プロトタイプ終了」の各項目が用意されている。

【0181】プロトタイプ実行モードにおいて設計者がシミュレーション時入力装置104(中のマウス)を操作して、PTMM1201から「プロトタイプ実行」を選択すると、PTM111はシミュレーション時入力装置104中のハードウェアデバイスとの通信を行い、ハードウェアデバイスから入力されたデータにより、当該PTM111を経由して対応するオブジェクトにイベントを与える。これによりプロトタイプ(シミュレーション)が実行される。

【0182】ここでは、画面上のソフトウェア部品がクリックされるなど、何らかのイベントが入力されると、該当する部品をリンク元とするリンクに設定されているメッセージがリンク先のソフトウェア部品あるいはカードに渡される。

【0183】一方、PTMM1201から「プロトタイプ一時停止」が選択されると、PTM111はハードウェアデバイスからの入力を受け付けられない状態に切り替え、プロトタイプ実行を一時中断する。この状態で再び「プロトタイプ実行」が選択されると、PTM111はハードウェアデバイスからの入力を受け付けられない状態から解除される。これにより、再びプロトタイプが実行される。

【0184】次に、PTMM1201から「プロトタイプ終了」が選択されると、PTM111からCLM10

9に終了が通知され、再び設計モードになる。

【0185】

【発明の効果】以上説明したように、本発明によれば、汎用のソフトウェア部品を用いて、GUI操作画面(ユーザインタフェース操作画面)のイメージ設計と動作の設計がGUI(グラフィカルユーザインタフェース)操作で容易に(プログラムを記述する必要なく)行うことができる。

【図面の簡単な説明】

10 【図1】本発明の実施形態に係るユーザインタフェース設計装置としてのコンピュータシステムの一例を示した図。

【図2】ユーザインタフェース設計装置の概略構成図。

【図3】図2のUI設計機構が立ち上げられたとき表示される、カードレイアウトマネージャメニュー(CLMM)、カードレイアウトウインドウ(CLMW)、ツールバーを含む画面イメージの一例を示した図。

【図4】マップビューアの表示例を示した図。

20 【図5】CLMが、サーバから供給されるOCXのインデックス情報(当該OCXの持つイベント、メッセージ、メソッド、属性(プロパティ)についての情報で、以下、イベント情報、メッセージ情報、メソッド情報、プロパティ情報と呼ぶ)を取得する処理動作を示したフローチャート。

【図6】テーブル114のインデックス情報の記憶例を示した図。

【図7】プロパティ情報の構成例を示した図。

【図8】イベント情報の構成例を示した図。

【図9】メッセージ情報の構成例を示した図。

30 【図10】引数情報の構成例を示した図。

【図11】設計中の画面イメージの具体例(電卓のGUI)を示した図。

【図12】カード上に配置するソフトウェア部品(OCX)の属性(プロパティ)を設定するために表示されるプロパティ設定ダイアログの一例を示した図。

【図13】リンク設定処理手順を説明するためのフローチャート。

【図14】リンクを設定する際に表示されるリンク設定ダイアログの一例を示した図。

40 【図15】リンク元のソフトウェア部品のイベントを設定する際に表示されるイベント選択リストの一例を示した図。

【図16】リンク先のソフトウェア部品のメッセージを設定する際に表示されるメッセージ選択リストの一例を示した図。

【図17】メッセージの引数を設定する際に処理手順を説明するためのフローチャート。

【図18】メッセージの引数を設定する際に表示される引数一覧ダイアログの一例を示した図。

50 【図19】メッセージの引数を設定する際に表示される

引数設定ダイアログの一例を示した図。

【図 20】ソフトウェア部品の属性（プロパティ）を引数の値に設定する場合、ソフトウェア部品を選択する際に表示される部品選択リストの一例を示した図。

【図 21】ソフトウェア部品の属性（プロパティ）を引数の値に設定する場合、属性（プロパティ）を選択する際に表示されるプロパティ選択リストの一例を示した図。

【図 22】複数のリンク元や複数のリンク先が選択されている場合のリンク設定を概念的に示した図。

【図 23】複数のリンク元や複数のリンク先が選択されている場合のリンク設定を行う際に表示されるリンク設定ダイアログの一例を示した図。

【図 24】プロトタイプ実行モードにおいて表示されるプロトタイプマネージャメニュー（PTMM）の一例を示した図。

【図 25】CLMにより出力装置 101 に表示される O C X のカスタマイズウインドウの一例として、例えば、部品名が「ボタン」である O C X のカスタマイズウインドウを示した図。

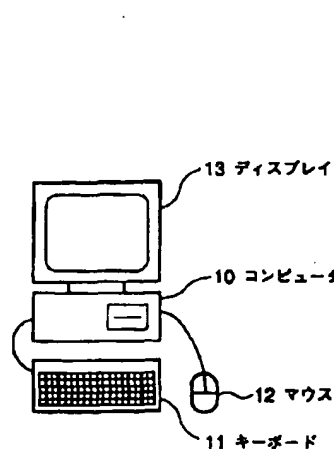
【図 26】計算結果表示箱という部品名の O C X の引数つきメッセージマクロを設定するページの一例を示した図。

【図 27】図 26 のメッセージマクロ設定ページ上に配置された新規作成ボタンを押下することにより表示される引数メッセージマクロ設定ウインドウの一例を示した図。

【図 28】計算結果表示箱という部品名の O C X のプロパティを変更するメッセージマクロを設定するページの一例を示した図。

【図 29】図 28 のプロパティ変更マクロ設定ページ上に配置された新規作成ボタンを押下することにより表示されるプロパティ変更のメッセージマクロ設定ウインドウの一例を示した図。

【図 1】



【図 3】



【図 30】アイテム定義ファイルに保存されるメッセージマクロ情報の構成例を示した図。

【図 31】アイテム定義ファイルに保存されるプロパティ変更マクロ情報の構成例を示した図。

【図 32】アイテム定義ファイルの一例を示した図。

【図 33】インデックス情報とアイテム定義ファイルとから作成されるテーブル 114 の記憶内容の一例を示した図。

【図 34】図 33 のテーブルに記憶されるプロパティ情報の構成例を示した図。

【図 35】図 33 のテーブルに記憶されるイベント情報の構成例を示した図。

【図 36】図 33 のテーブルに記憶されるメッセージ情報の構成例を示した図。

【図 37】メッセージ選択リストの作成および表示処理動作を説明するためのフローチャート。

【符号の説明】

101…出力装置

102…設計時データ入力装置

20 103…ユーザインタフェース設計機構（UI 設計機構）

104…入力装置

105…データベース（DB）

106…メモリ

107…エディティング部

108…プロトタイプ実行部

109…カードレイアウトマネージャ（CLM）

111…プロトタイプマネージャ（PTM）

112…サーバ

30 113…レジストリ

114…テーブル

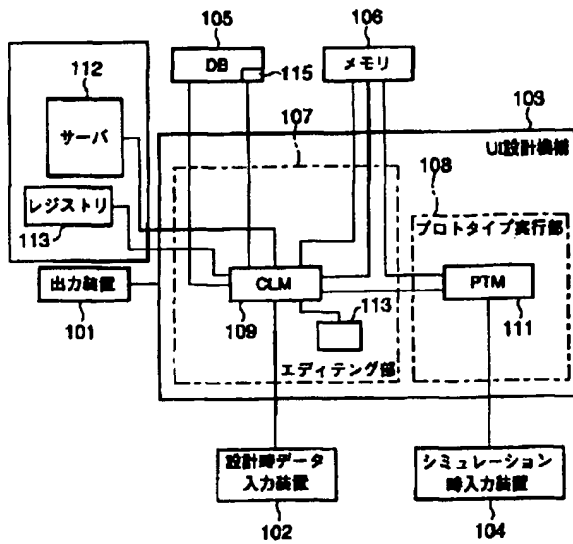
115…アイテム定義ファイル

204…ツールバー

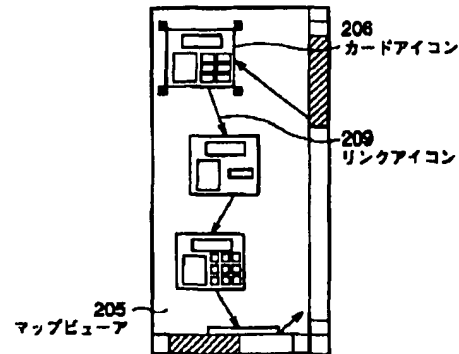
【図 7】

プロパティ情報	
CLSID	OCXのクラスID
MEMBERID	メンバID
NAME	プロパティ名
VARTYPE	プロパティの型

【図2】



【図4】



【図21】

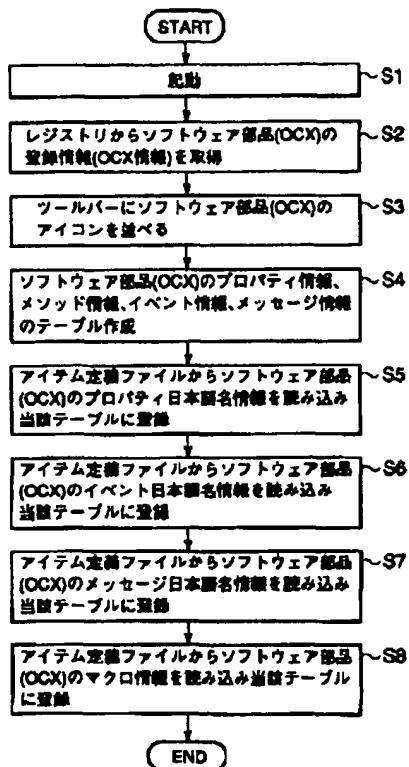
プロパティ名選択リスト
データ(表示文字)

【図15】

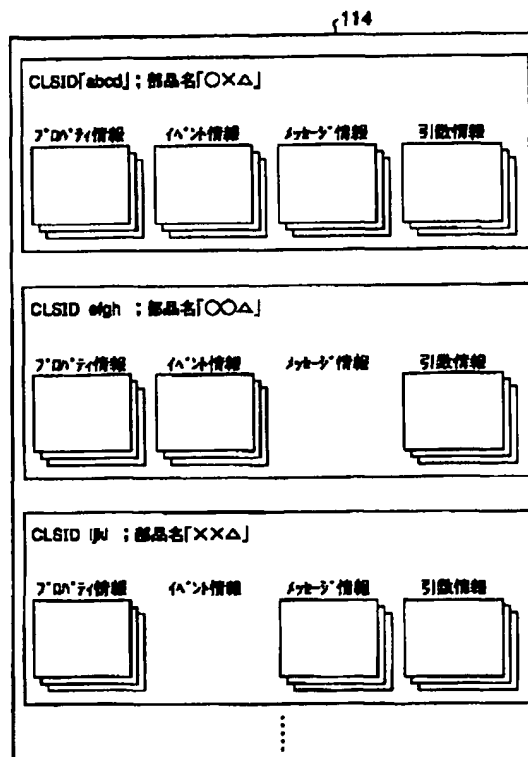
イベント選択リスト

ボタンダウン
ボタンアップ
クリック
ムーブ

【図5】



【図6】



【図16】

メッセージ選択リスト

数値入力
加算
減算
乗算
除算
計算

【図20】

部品名選択リスト

リンク元
ボタンB1
ボタンB2

【図8】

イベント情報	
CLSID	OCXのクラスID
MEMBERID	メンバID
NAME	イベント名
ARGS	引数リスト(引数名,...)

【図9】

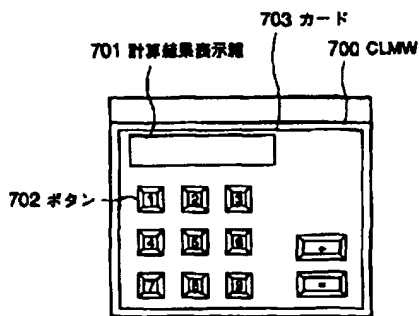
メッセージ情報	
CLSID	OCXのクラスID
MEMBERID	メンバID
NAME	メッセージ名
ARGS	引数リスト(引数名,...)
RETURNINFO	戻り値情報

【図10】

引数情報	
引数ID	
引数名	
型	

【図14】

【図11】



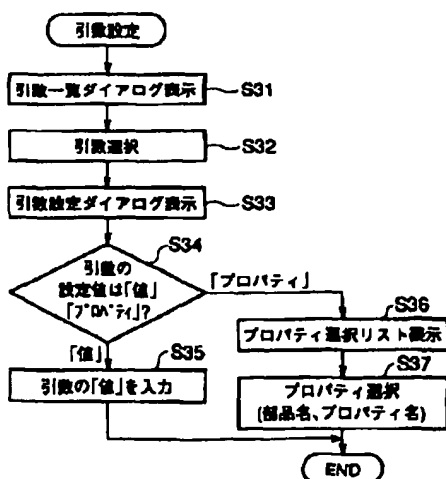
【図12】

Figure 12 is a 'プロパティ設定ダイアログ' (Property Setting Dialog). It contains two sections. The first section has 'プロパティ名' (Property Name) set to 'データ' (Data), '型' (Type) set to 'int', and '設定値' (Setting Value) set to '1'. The second section has 'プロパティ名' set to '文字色' (Text Color), '型' set to 'null', and '設定値' set to '黒' (Black).

Figure 14 is a 'リンク設定ダイアログ' (Link Setting Dialog). It includes fields for 'リンク名' (Link Name) set to '数字代入' (Numeric Substitution), 'リンク元' (Link Source) set to 'ボタン1' (Button 1), 'イベント' (Event) set to 'ボタンクリック' (Button Click), 'リンク先' (Link Destination) set to '計算結果表示欄' (Calculation Result Display Area), and 'メッセージ' (Message) set to '数字入力' (Numeric Input). There are 'OK' and 'CANCEL' buttons at the bottom.

900...リンク設定ダイアログ
 901...リンク名設定エディットボックス
 902...リンク元表示エディットボックス
 903...イベント選択領域
 904...リンク先表示エディットボックス
 905...メッセージ選択領域
 906...リンク設定ボタン
 907...OKボタン

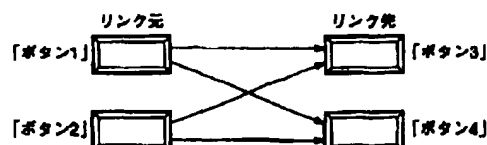
【図17】



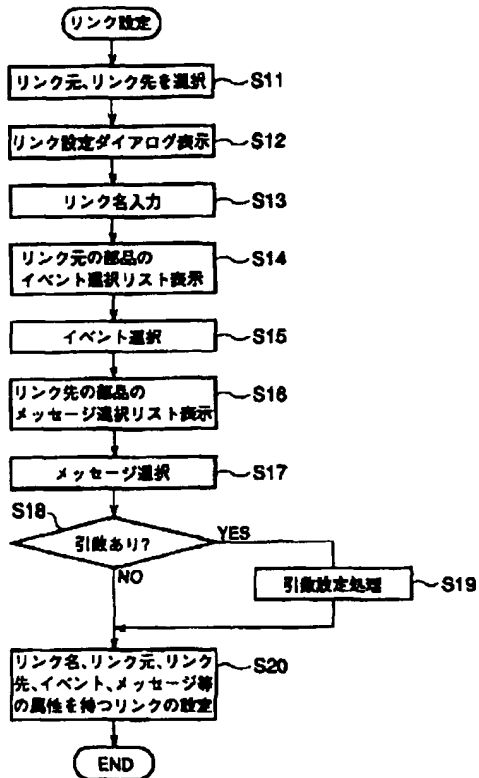
【図18】

Figure 18 is an '引数一覧ダイアログ' (Argument List Dialog). It shows 'メッセージ名' (Message Name) as '数字入力' (Numeric Input). Below, there is a list of arguments with '1' and '数字' (Numeric) selected. A '設定' (Setting) button is next to the list. At the bottom are 'OK' and 'CANCEL' buttons.

【図22】



【図13】



【図19】

引数設定ダイアログ

引数名: 数字 (1104 引数名表示フィールド)

型: int (1105 引数型表示フィールド)

設定値: 0 (1106 設定値切替スイッチ, 1107 設定値入力フィールド)

プロパティ: ☒ プロパティ (1108 部品名選択領域)

部品名: リンク元 (1109 プロパティ名選択領域)

プロパティ名: Data

OK CANCEL (1100)

【図30】

メッセージマクロ情報	
MACRONAME	マクロ名
LINKSRC	リンク先のOCX種類
MESNAME	メッセージ名
UNIT	メッセージの引数の個数
ARGINFO*	引数情報テーブルのポインタ

【図34】

プロパティ情報	
CLSID	OCXのクラスID
MEMBERID	メンバID
NAME	プロパティ名
VARTYPE	プロパティの型
JNAME	日本語名

【図23】

リンク設定ダイアログ

リンク名: 数字代入 (901)

リンク元: 数字1, 数字2 (902)

イベント: 数字リンク (903)

リンク先: 数字3, 数字4 (904)

メッセージ: 数字入力 (905), 引数 (906)

OK CANCEL (907)

【図24】

PTMM		
プロトタイプ実行	プロトタイプ一時停止	プロトタイプ終了

【図26】

計算結果表示用のカスタマイズウインドウ

タブ: プロパティ, イベント, メッセージ

引数つきメッセージマクロ, プロパティ変更マクロ

マクロ名: 計算, 計算結果表示, 計算結果表示 (1500 マクロ名リスト)

新規作成 (1501), 削除 (1502)

OK CANCEL

【図35】

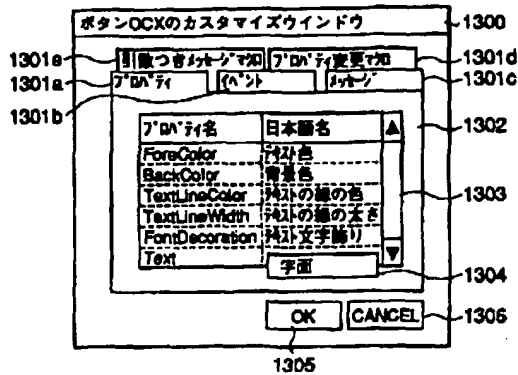
イベント情報	
CLSID	OCXのクラスID
MEMBERID	メンバID
NAME	イベント名
ARGS	引数リスト(引数名、...)
JNAME	日本語名

【図36】

メッセージ情報	
CLSID	OCXのクラスID
MEMBERID	メンバID
NAME	メッセージ名
ARGS	引数リスト(引数名、...)
RETURNINFO	戻り値情報
JNAME	日本語名

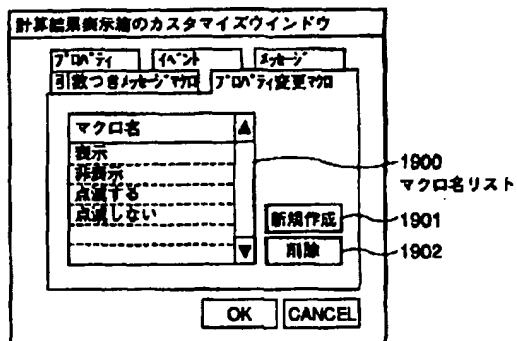
900...リンク設定ダイアログ
 901...リンク名設定エディットボックス
 902...リンク元表示エディットボックス
 903...イベント選択領域
 904...リンク先表示エディットボックス
 905...メッセージ選択領域
 906...引数設定ボタン
 907...OKボタン

【図25】



- 1300: カスタマイズウィンドウ
 1301: タブ
 1302: ページ
 1303: プロパティリストボックス
 1304: テキスト入力ボックス

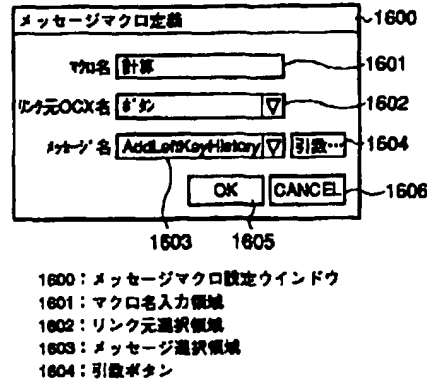
【図28】



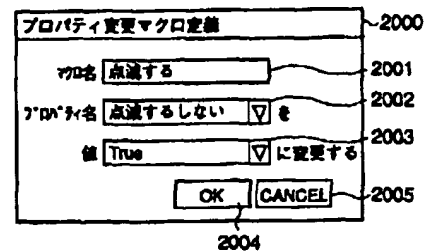
【図31】

プロパティ変更マクロ情報	
MACRONAME	マクロ名
PROPNAME	プロパティ名
VARTYPE	値

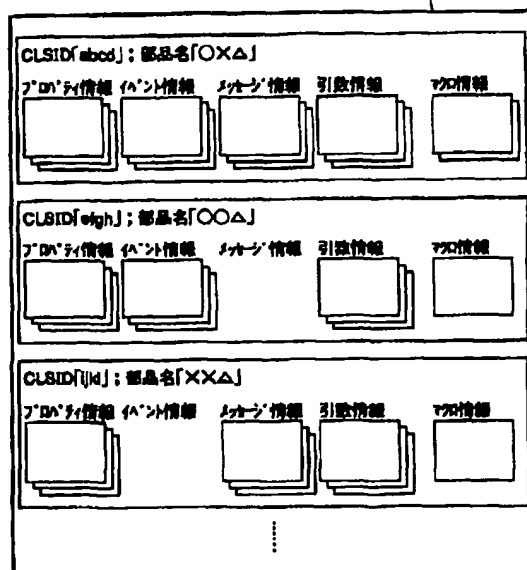
【図27】



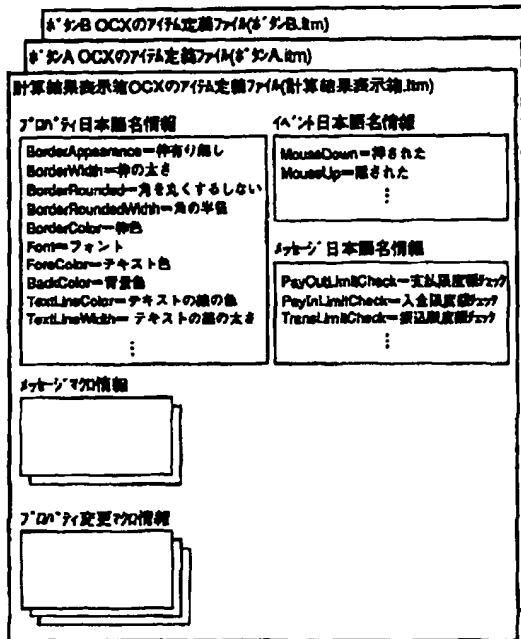
【図29】



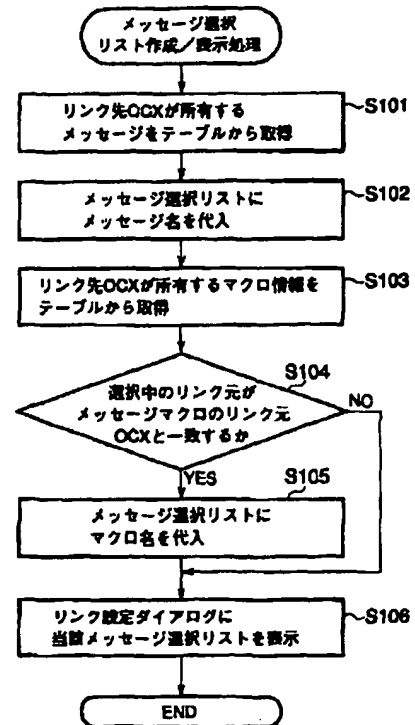
【図33】



【図 32】



【図 37】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ ~~FADED~~ TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.